MICROCOMPUTADOR CURSO PRÁTICO

TIRE O MÁXIMO DO MICRO EM 30 SEMANAS

PUBLICAÇÃO SEMANAL ILUSTRADA

26

Cr$ 12.900

Livro eletrônico • Robôs na educação
Biorritmo por computador • Pacotes de software

rioGráfica

MANAUS, ALTAMIRA, MACAPÁ, JI-PARANÁ, PORTO VELHO, RIO BRANCO, SANTARÉM, BOA VISTA, SINOP, ALTA FLORESTA: Cr$ 16.770

MICROCOMPUTADOR CURSO PRÁTICO

APLICAÇÕES

Enciclopédia eletrônica 517

Busca inteligente 529

CHIPS & BYTES

Nova aprendizagem 520

Biorritmo 528

SOFTWARE

Pacotes financeiros 523

RAIO X

Expert 525

LINGUAGENS

Rotinas genéricas 532

PERFIL

Edisa 535

TESTE

Depois de ler com atenção todo o fascículo, responda a estas perguntas. Veja a solução no próximo número, junto com um novo teste.

1)Qual a empresa estatal brasileira encarregada de transmitir on line os bancos de dados de acesso público?

2)A linguagem C-LOGO, ou LOGO CONCORRENTES, apresenta dois novos comandos — um para a execução indefinida de uma rotina e outro para a execução condicional de uma função. Quais são esses comandos?

3)Qual o motivo de haver duas versões distintas da linguagem ASSEMBLY?

4)Qual o comando equivalente ao GOSUB do BASIC na linguagem ASSEMBLY, em suas duas versões?

RESPOSTAS DO TESTE ANTERIOR

1) CAD significa "Computer Aided Design", ou seja, "projeto com auxílio do computador".

2) Os robôs "ouvem" por meio de microfones e software para reconhecimento de som e fala.

3) A ciência que procura projetar instrumentos e ambientes de trabalho agradáveis e adequados ao ser humano é a ergonomia.

4) Os usuários de micros consideram a cor âmbar nas telas a mais agradável à vista.

PLANO DA OBRA

MICROCOMPUTADOR - CURSO PRÁTICO é uma coleção de 30 fascículos de periodicidade semanal. Os 29 primeiros terão, cada um, vinte páginas internas (miolo) e quatro capas, além de uma introdução de dezesseis páginas acompanhando o fascículo 1. O fascículo 30 incluirá o índice geral da obra. Os miolos, encadernados, constituem dois volumes de um curso prático de microcomputação, com seções indicadas por tarjas de diferentes cores.
As duas primeiras capas são descartáveis; as 3.as e 4.as trazem programas de jogos. Encadernadas, formam um volume de cem páginas.

COMO ENCADERNAR

As capas duras, incluindo as guardas, estarão à venda simultaneamente com os fascículos 10 (**Volume 1**) e 30 (**Volume 2** e **Jogos**).
Volume 1 - Deve ser encadernado com os elementos dispostos nesta ordem: guardas; frontispício e introdução (publicados com o fascículo 1); miolos dos fascículos de 1 a 15; e guardas.
Volume 2 - Guardas; frontispício (que virá com o fascículo 30); miolos dos fascículos de 16 a 30; e guardas.
Jogos - Guardas; frontispício e sumário (que constituem as 3.as e 4.as capas do fascículo 30); 3.as e 4.as capas dos fascículos de 1 a 29; e guardas.

rioGráfica

CONSELHO DE ADMINISTRAÇÃO: Roberto Irineu Marinho; João Roberto Marinho; José Roberto Marinho; Oscar Neves; **Diretoria:** Oscar Neves; Danilo Esteves Costa; João J. Noro; Marcos March; Nilo Sérgio de Almeida; **GRUPO EDUCAÇÃO E CULTURA - Divisão Comercial - Diretor:** Jaime Rodrigues; **Gerente de Planejamento:** João de Azeredo Coutinho; **Gerente de Vendas:** Rubens Barbosa; **Coordenadores de Promoção:** Paulo Cesar M. Seixas; Edgard Mello M. Neto; **Gerente de Circulação:** Norberto Martin; **Supervisor de Operações:** Abel Pereira Pinto; **Editorial - Diretor:** João J. Noro; **Editor-chefe:** Mauricio Rittner; **Diretor de Arte:** Adalberto Cornavaca; **Administração Editorial:** Heitor de Souza Paixão; **Editor:** Jesse Navarro; **Editor-assistente:** Claudio Marcondes; **Coordenadora de Revisão:** Isis Augusta Loyolla; **Revisores:** Maria da Graça Mendonça Couto; Sérgio Roberto Torres; **Chefe de Arte:** Bonifácio Duardes Miranda; **Chefe de Estúdio:** José Yuji Kuribayashi; **Assistentes de Arte:** Miguel Luis Escámez Simón; Edemilson Ribeiro da Silva; Wagner Celestino; **Colaboradores - Editores-assistentes:** Isa Mara Lando; Luiz Carlos Pizarro Marin; **Texto:** Carlos Eduardo Silveira Matos; José Antônio Arantes; Jussara Maturo; Paulo Brito; Solange A. Menezes; **Preparação de Texto:** Luiz Carlos Cardoso; **Tradução:** Luiz Roberto de Godoy Vidal; Maria Clara Cescato; Maria Ester Menezes Martinho; Fernando Dorfman Knijnik; Newton Roberval Eichemberg; **Pesquisa:** Stella Maria Quentel; **Assessoria Técnica:** Geraldo Coen; Gustavo José Ferreira Grubba; José Rubens Salles Toledo; Lauro de Lauro; Pierluigi Piazzi; **Estúdio:** Vagner Facuri de Oliveira. As fotos não creditadas pertencem à obra original; Gradiente; Edisa. © APSIF Copenhagen, 1984; © Orbis Publishing Ltd., 1984; © Editora Rio Gráfica Ltda., 1985, para a língua portuguesa. **Composição:** Editora Rio Gráfica Ltda. e AM Produções Gráficas Ltda. **Impressão:** JBIG.

NÚMEROS ATRASADOS - Você poderá comprar os exemplares que faltam em sua coleção, pelo preço do último fascículo posto à venda, em sua banca de jornal preferida. Caso não consiga obtê-los, faça sua solicitação por carta endereçada diretamente à Editora Rio Gráfica Ltda., rua Itapiru, 1209, Rio Comprido, Rio de Janeiro, CEP 20251. O atendimento será feito por via postal. Nas cidades do Rio de Janeiro e de São Paulo, as compras poderão ser efetuadas pessoalmente nos seguintes endereços: Rio de Janeiro - Rua Itapiru, 1209, Rio Comprido; São Paulo - Rua Frei Caneca, 1152, Consolação.

Distribuidor exclusivo para todo o Brasil: Fernando Chinaglia - Rua Teodoro da Silva, 907, Rio de Janeiro. Distribuidor para Portugal: Electroliber Lda. - Rua Prof. Reinaldo dos Santos, 1488, Lisboa.

ENCICLOPÉDIA ELETRÔNICA

Uma das maiores vantagens do uso dos computadores na educação é o acesso a bancos de dados grandes e sempre atualizados. Examinaremos alguns programas especificamente elaborados para os jovens alunos.

Há muitos benefícios educacionais em se obterem informações de uma "biblioteca eletrônica"; por exemplo, acesso imediato, sem que seja preciso sair da sala, nem mesmo da carteira, para conseguir os dados. Isso elimina o problema de alunos que "não têm o livro". Além disso, registros computadorizados são facilmente atualizáveis, enquanto as informações em livros de texto não — só poderão ser alteradas numa eventual reimpressão. Os custos da produção de livros representam sério problema ao fornecimento de material didático, e as escolas com freqüência se vêem forçadas a adotar publicações desatualizadas. O pior é que o alto custo do texto impresso também impede a expansão de novos métodos educacionais. Por outro lado, uma escola equipada com computadores requer, para a alteração dos currículos, apenas a troca dos discos mestres.

Os benefícios educacionais evidenciam-se particularmente na combinação de microcomputadores e recursos de telecomunicação para serviços de bancos de dados on line (conexão direta). Ter, por exemplo, toda uma enciclopédia em discos constituiria trunfo extremamente útil. Se pudermos encontrar a informação desejada pesquisando nos índices alfabéticos comuns, será perfeitamente adequado pegar um exemplar do livro da estante. Mas, se por exemplo quisermos conhecer todos os marsupiais que vivem

O mundo que as rodeia
A despeito do receio de que o uso de computadores nas escolas confinaria as crianças ao estreito mundo das telas de vídeo, a prática tem mostrado que o software de banco de dados, em particular, pode conduzir a todo tipo de atividades "extracomputador". A necessidade de coletar informações para alimentar o banco de dados pode ser abordada de muitas maneiras diferentes, inclusive a do "passeio pela natureza", onde os alunos são encorajados a observar, registrar e classificar os elementos do meio ambiente.

em tocas, ou listar todos os países que aplicam determinada porcentagem do PNB na educação, vai ser muito cansativo pesquisar e compilar tudo à mão.

Entretanto, uma enciclopédia com, digamos, 26 volumes exigiria, para seu armazenamento, face aos recursos de pesquisa sistemática, uma capacidade imensa de processamento; é aí que os serviços on line entram em ação. Pode-se acessar esse sistema com um modem padrão de 300 ou 1.200 bauds e uma linha telefônica, ficando disponível uma gama vastíssima de informações.

No Brasil, o Serpro e o IBGE colocam seus bancos de dados à disposição do usuário através da Embratel. O Dialog, nos Estados Unidos, é o maior serviço de banco de dados on line do mundo e um dos mais utilizados no campo da educação. Contém duzentos bancos, com mais de cem publicações e 100 milhões de itens de informação — compêndios de todos os assuntos, de história da arte a zoologia. O Dialog fornece, além do autor e título dos documentos ou trabalhos relevantes, também um resumo com uma descrição substancial do artigo — não raro contendo toda a informação de que se necessita. Esta pode então ser impressa ou solicitada como listagem, a um custo bastante razoável. Os bancos de dados disponíveis estão sendo complementados por uma crescente quantidade de software de pesquisa, para ajudar os usuários a extrair e classificar as informações neles contidas.

O banco de dados da firma inglesa Prestel apresenta um tipo diferente de abordagem — o Schools Link ("ligação escolar") —, oferecendo às escolas informações sobre todos os aspectos da computação aplicada à educação, além de proporcionar contatos e debates. Destina-se a professores e contém resenhas de software, de robôs educacionais, de livros e sugestões para projetos; também permite que o equipamento do professor ou da escola acesse programas educacionais.

Entretanto, além da "biblioteca eletrônica", os bancos de dados estão se tornando populares nas salas de aula, aí desempenhando papel ativo. Já existem vários programas que proporcionam aos alunos a experiência de manipular informações e estruturá-las de acordo com suas próprias necessidades.

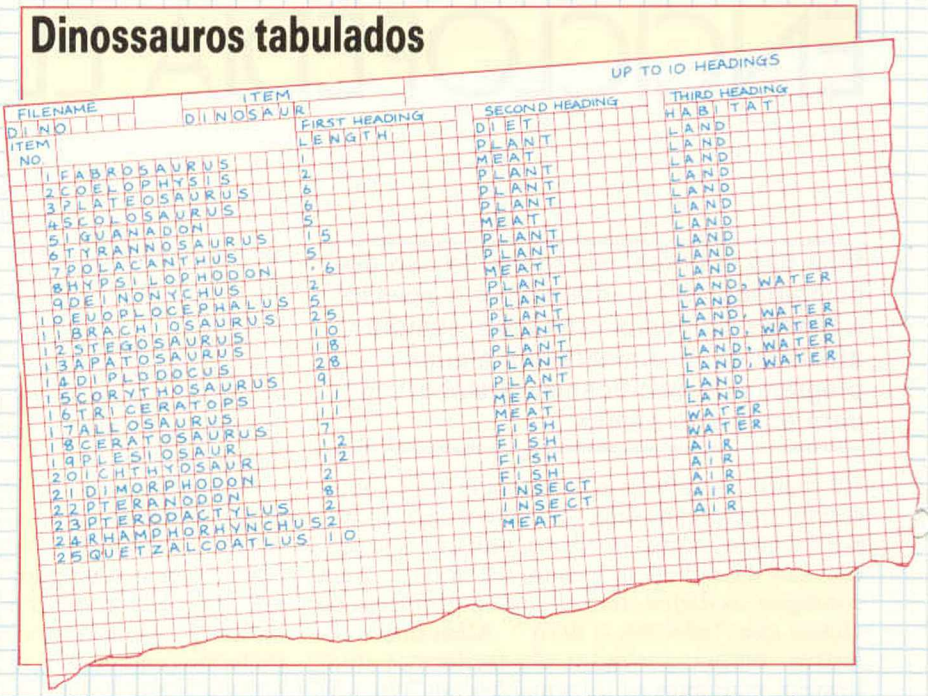
O banco de dados Factfile

Numa conferência educacional, na universidade de Cambridge, em 1981, discutiu-se a possibilidade de uso dos bancos de dados. Esse debate inspirou a criação de um programa chamado Factfile ("Arquivo de Fatos"), que permite a crianças, mesmo pequenas, montar arquivos de dados sobre qualquer tópico que escolham.

O papel de um banco de dados como o Factfile no estímulo à descoberta e aprendizado ativo logo se torna evidente. Suponhamos que as crianças queiram criar um banco de dados contendo informações sobre seus professores. Primeiramente, terão de decidir as categorias nas quais organizarão as informações. Isso dependerá do que já sabem e do que conseguem descobrir sozinhas. Podem, por exemplo, incluir sexo, idade, altura, disciplina lecionada e "tipo de professor". Determinada a estrutura do banco, deverão coletar informações para cada categoria. Nesse ponto, o exercício se transforma numa operação educacional diversificada — no mínimo, as crianças terão de coletar informações físicas. Se forem necessárias medições, terão de determinar as unidades de medida e os dados a incluir. O processo vai se tornando mais complexo à medida que as crianças começam a entender o significado das categorias criadas por elas. O campo "tipo de professor" requer bastante discussão. O que significa: "Que disciplina leciona?" ou "É um professor legal ou chato?".

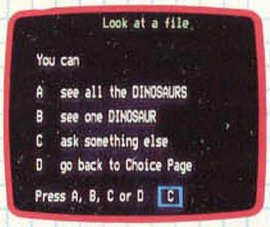
Encerrada a compilação do banco de dados, outras crianças poderão dirigir-lhe perguntas, correlacionando informações de várias categorias. Talvez queiram listar todas as professoras classificadas na categoria "legal" ou todos os professores de matemática do sexo masculino. E o processo de aprendizagem avança mais um degrau quando as crianças começam a analisar os resultados. O Factfile permite que as crianças gravem seu banco de dados em disco ou cassete e acrescentem mais informações posteriormente, podendo retornar aos arquivos de dados sempre que necessário.

Uma das críticas ao uso de computadores por crianças é que estas se sentam frente a um vídeo, executando atividades sem relevância para o mundo em que vivem. Mas, além de dar às crianças uma experiência direta com bancos de dados, o Factfile leva a inúmeras atividades não computacionais. Ao criar um arquivo com informações sobre a flora local, as crianças teriam de registrar cuidadosamente os detalhes sobre ca-



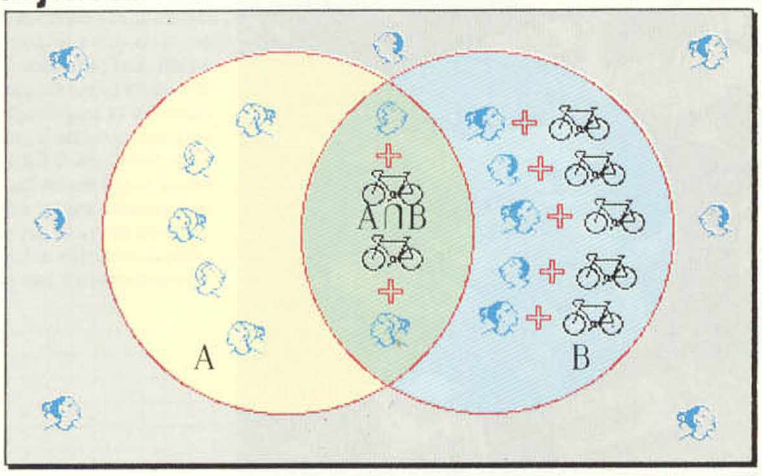
A memória permanece
O Factfile proporciona excelente introdução ao gerenciamento de dados. Inclui um arquivo de demonstração chamado Dino, contendo informações sobre dinossauros (alimentação, habitat etc.). As crianças são estimuladas a familiarizar-se com o Dino e a catalogar o conteúdo do arquivo (como acima), antes de passarem à criação de seus próprios bancos de dados.

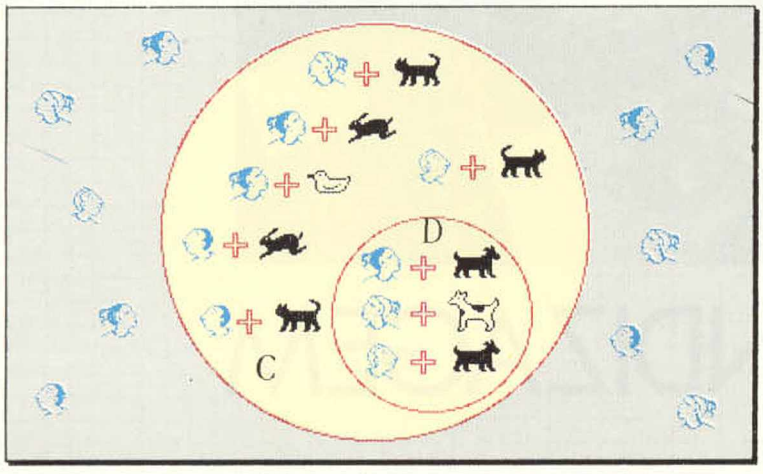
Pedido de pesquisa



Menu do Dino

Conjuntos





da planta e, então, desenvolver seu próprio sistema de classificação que lhes permitisse montar o banco de dados.

Há um outro pacote popular, concebido para crianças menores, o Your Facts (''Seus Fatos''). Pergunta-se à criança seu nome, se é menino ou menina, se tem algum animal de estimação, relógio de pulso, bicicleta, irmãos ou irmãs. Quando as informações sobre uma criança terminam de ser introduzidas, o programa pergunta se alguém mais gostaria de entrar com seus dados. Há espaço para fatos sobre quarenta crianças e, assim, pode-se incluir uma classe inteira. Quando todas as informações tiverem sido introduzidas e ''Não'' for digitado em resposta a ''Alguém mais gostaria de entrar com seus dados?'', as crianças retornam ao menu. Podem então ver todos os registros, verificar quais delas caíram em determinada categoria — como a das que têm relógio ou a das que têm bicicleta — ou podem brincar com um videojogo. Neste, o computador tenta acertar o nome da criança, depois de fazer perguntas como ''Você é um menino?'', ''Você tem relógio?'', e assim por diante. Então pergunta, por exemplo, ''Seu nome é Tiago?''. O Your Facts revela-se uma introdução ideal, para crianças pequenas, ao conceito de banco de dados. Refere-se a elas pessoalmente, encoraja a leitura e a escrita e é um exercício divertido.

Informação classificada

A construção e a pesquisa em bancos de dados relacionam-se a outros trabalhos feitos por crianças nas escolas.

O programa de matemática moderna introduz a idéia de conjuntos desde tenra idade. A teoria dos conjuntos diz respeito à classificação de dados e às relações entre cada classificação. Montar um banco de dados é uma atividade análoga a essa, e consiste em classificar as propriedades de um item em campos e registros, de forma que o banco possa ser pesquisado para fornecer as relações entre os itens.

Em particular, os conceitos de interseção e subconjunto têm importante significado na pesquisa do banco de dados, correspondendo à criação de sinopses. A interseção é a sobreposição de dois conjuntos — por exemplo, a interseção do conjunto de crianças louras com o de crianças que têm bicicletas é o conjunto de crianças louras que têm bicicletas.

O programa One World

Uma empresa australiana produtora de software, a Active Learning Systems, produziu um extenso banco de dados com informações sobre quase todos os países do mundo. O programa chama-se One World (''Um Único Mundo'') e inclui indicações aos professores e planilhas para a classe e para os alunos individualmente. Contém mais de trinta itens de informação sobre cada país, incluindo fatos a respeito do sistema político, importações, exportações, línguas, limites, religiões, taxa de alfabetização, fatos históricos e tratados. Dá as porcentagens das populações rural e urbana, assim como a porcentagem da força de trabalho nos setores de produção, manufatura e serviços. Também fornece as proporções de áreas desérticas, cultivadas e florestais. O que torna o programa tão atraente, como ocorre com a maioria dos bancos de dados, são seus recursos de pesquisa e análise dos dados.

O menu principal dá uma série de opções:

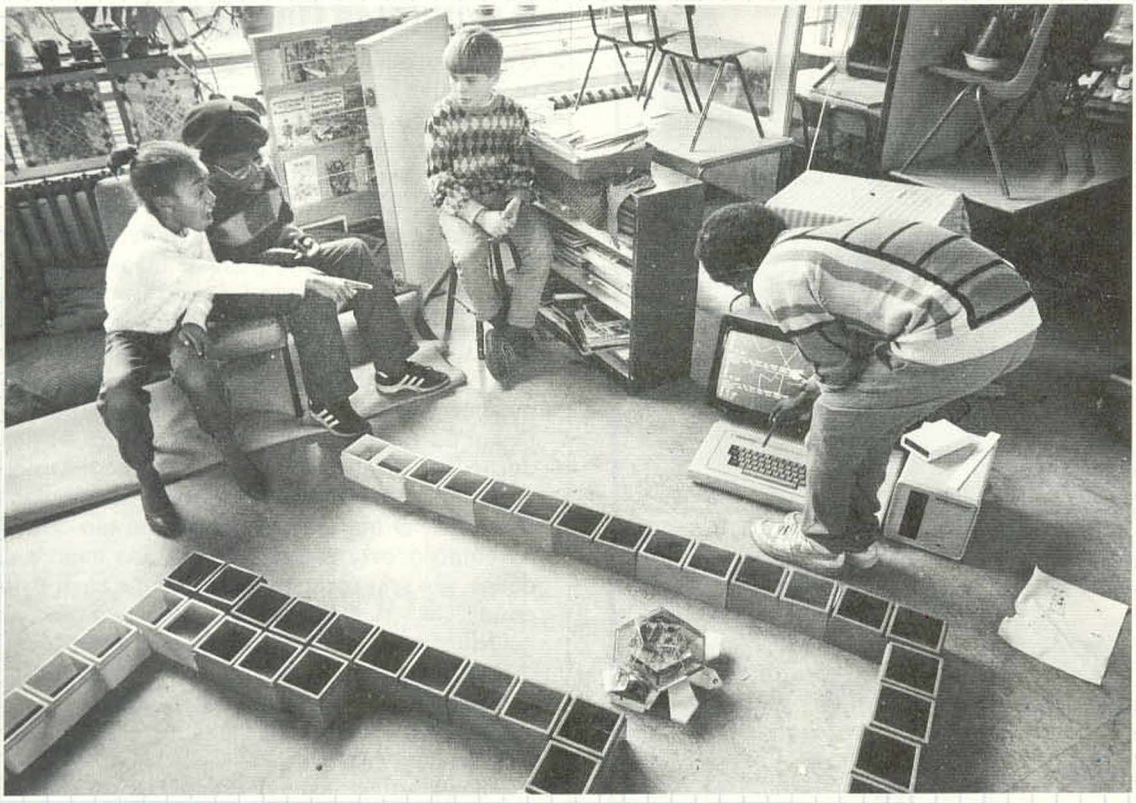
1. Apresentar um país
2. Pesquisar com informações exclusivas
3. Analisar usando vários critérios
4. Auxílio à consulta
5. Saída

Se uma das crianças conhece um certo Mitterrand, um líder político europeu, e quer encontrar seu país, escolhe a segunda opção. O menu Dados de pesquisa permitiria à criança introduzir o nome Mitterrand em 3. Líder do governo e 4. Chefe de Estado. O programa então informaria que François Mitterrand é o chefe de Estado da França.

A opção Análise é o recurso mais poderoso do programa. Permite às crianças analisarem as informações, usando até três opções, a partir de vinte critérios. Por exemplo, poderiam determinar em quais países europeus a maioria da população vive no campo. Isso exigiria somente duas opções: região e porcentagem da população. Se as crianças ouvissem que os mineiros de bauxita no Suriname tinham entrado em greve, rapidamente descobririam a localização desse país, saberiam que a bauxita é seu principal produto de exportação e que a greve poderia ter efeito devastador sobre a economia.

O One World desenvolve a capacidade de pesquisa e interpretação de dados, encorajando as crianças a tirarem conclusões a partir da análise destes. É um grande apoio para as aulas de história, geografia e ciências sociais.

Um banco de dados fornece às crianças boa parte dos dados disponíveis e as estimula a pesquisar as respostas. Elas podem observar como o computador manipula as informações, entender a importância de fazer as perguntas corretas e organizar adequadamente suas entradas. Aquelas que, após experimentar o uso de um banco de dados, mais tarde entrarem para cursos superiores e para o mercado de trabalho certamente já possuirão importantes e relevantes habilidades.



Iniciação à geometria
Neste exercício os alunos iniciam-se nos princípios da geometria com a tartaruga Valiant. Eles participam ativamente na construção do labirinto e na programação da tartaruga, para fazê-la caminhar pelos obstáculos. O trabalho ensina os rudimentos do deslocamento angular e outros conceitos abstratos que as crianças costumam achar incompreensíveis e sem graça.

NOVA APRENDIZAGEM

Uma tartaruga mecânica auxilia o jovem estudante, na sala de aula, a entender conceitos abstratos de matemática. Além de divertidos, os robôs podem representar um eficiente método de ensino.

O uso de robôs na educação é uma tentativa de solucionar o problema — premente nas crianças pequenas — da compreensão de conceitos abstratos sem o correspondente modelo concreto. Complexas noções de trigonometria, por exemplo, podem ser demonstradas fisicamente; representam-se os processos mentais pelos movimentos reais de uma tartaruga tridimensional.

A geometria sempre foi ensinada com instrumentos tradicionais — lápis, régua e transferidor. Para uma criança de oito anos, aprender como medir um ângulo tem pouca importância, muito menos compreender os conceitos mais amplos da geometria. Por outro lado, os ângulos ganham um significado muito maior quando determinam se uma tartaruga, programada pelo aluno, irá ou não se chocar com uma floresta de garrafas plásticas dispostas no chão da sala de aula. Ainda que a geometria permaneça nebulosa, haverá uma motivação para entender suas aplicações práticas. A diferença entre distâncias de 15 e 20 cm fica rapidamente evidenciada quando resulta de uma escolha entre um ângulo de 5 ou 8 graus.

Um dos mais importantes aspectos da tecnologia das tartarugas está na possibilidade de aprendizagem através de tentativa e erro. A criança modifica a programação até conseguir um traçado satisfatório, enquanto vai adquirindo a compreensão das propriedades dos ângulos, retas e medidas. A ocorrência de um erro significa apenas que algo mais precisa ser tentado, até que ocorra o movimento desejado, não que algo tenha fracassado.

A programação de um braço mecânico para levantar e mover um objeto requer um raciocínio nos três planos do movimento; imaginar esse deslocamento no papel, que é bidimensional, não se mostra muito fácil. Com um braço robótico, no entanto, os planos do movimento são visualizados com clareza. Os resultados da programação aparecem, facilmente se detectam seus erros, e as alterações apropriadas podem ser introduzidas até que a mancha transcorra perfeita. O robô fornece uma realimentação instantânea ao programador, demonstrando cada ângulo e cada medida em seqüência, de maneira muito mais eficiente que um exercício tradicional com lápis e papel. Isso também permite ao aluno avaliar seu próprio trabalho.

Exemplo concreto

As crianças freqüentemente se mostram incapazes de compreender conceitos abstratos, mesmo quando apresentados com uma demonstração física dos princípios envolvidos. Um exemplo bem conhecido é o da conservação de quantidades e relações transitivas, em que se coloca a criança diante de recipientes com formas diferentes, mas de igual capacidade. Neste exemplo, A, B e C contêm quantidades iguais de líquido, enquanto B e B' são idênticos.

O instrutor transfere o conteúdo de A para B' e solicita à criança que verifique a igualdade de níveis em B e B'. E retorna, então, o líquido de B' para A.

Em seguida, ele executa operação idêntica com os recipientes C e B'.

A e C são colocados lado a lado e pergunta-se à criança se algum deles está com mais líquido que o outro, ou se ambos contêm a mesma quantidade.

Interroga-se à criança se A, B e C contêm a mesma quantidade de líquido. Nas duas últimas etapas da experiência, se a resposta foi correta, pede-se que ela explique seu raciocínio. Crianças com menos de sete anos em geral são incapazes de compreender os princípios envolvidos ou de responder com acerto. Essa é uma das experiências aplicadas pelo psicólogo suíço Piaget, cujo trabalho influenciou o desenvolvimento do LOGO por Papert. O uso de robôs na educação apresenta conceitos abstratos à criança, dando-lhe um modelo concreto e um papel mais ativo na aprendizagem.

Um brinquedo robótico popular em escolas infantis (de jardins de infância à primeira série) é um tanque futurístico chamado Big Trak, dotado de uma pequena memória e um teclado na parte superior. Digitando-se as setas e mais um número de 1 a 99, comandam-se os movimentos do tanque para todas as direções. As teclas numéricas permitem ainda indicar a distância a ser percorrida. Podem ser armazenados até oito comandos de cada vez. Utiliza-se esse brinquedo como substituto da tartaruga, encorajando as crianças a programarem rotinas simples. O Big Trak demonstra que os comandos são executados na ordem em que foram dados — uma útil iniciação à programação de computadores.

Objeto que ajuda a pensar

Embora a tartaruga possa auxiliar no ensino da matemática, essa não é sua principal função; ela foi projetada para ensinar a criança a programar um computador. O idealizador da tartaruga, Seymour Papert, acha que as crianças devem programar o computador, em lugar de serem por ele programadas. Essa filosofia é evidente na linguagem LOGO, desenvolvida por Papert para fins educacionais. Ele descreve a tartaruga como "um objeto para pensar", com o qual uma criança aprende os princípios básicos da programação utilizando a linguagem LOGO para controlar as ações do robô.

Enquanto a maioria dos projetistas se concentrava no desenvolvimento de hardware, Paul Cheung, da Universidade de Edinburg, sentiu a necessidade de um software mais versátil, e desenvolveu os aspectos robóticos do LOGO. Cheung desejava que as crianças pudessem controlar uma variedade de robôs. O LOGO é ideal para programar tartarugas, mas se mostra limitado para programar outros dispositivos.

Tomando o LOGO como ponto de partida, Cheung reestruturou a linguagem, permitindo o controle de mais funções e aumentando as possibilidades de receber sinais de entrada a partir de vários dispositivos, como sensores de toque, leitoras de código de barras e fotossensores. O resultado foi o Concurrent-LOGO, ou C-LOGO, uma linguagem capaz de realizar processamento "paralelo", isto é, executar diversas funções simultaneamente. O C-LOGO pode controlar até oito processadores de cada vez.

Duas funções particularmente interessantes são FOREVER ("para sempre") e WHENEVER ("sempre que"). A primeira, FOREVER, inicia um determinado processo que só se encerra pelo C-LOGO. A WHENEVER aguarda que surja uma determinada condição e, então, aciona uma função específica; é essencial para o processamento paralelo. O C-LOGO permite ligar e desligar chaves e verificar se alguma está ligada ou desligada. Utiliza-se esse recurso com o sensor — se um sensor de toque encosta num obstáculo, liga uma chave, informando ao computador que o contato foi realizado. Uma criança, utilizando a função WHENEVER, pode programar um

carrinho para evitar obstáculos, ordenando o desvio do robô-carrinho, cada vez que houver contato com algum obstáculo. O C-LOGO é mais adequado para os alunos do segundo grau, onde Paul Cheung o utilizou com diversos dispositivos de controle.

Alunos de treze a quinze anos desenvolveram programas para acionar os botões que controlam os movimentos das portas e janelas de uma casa de brinquedo automatizada. Essa casa tinha alarme contra roubos, ativado pela abertura de uma janela, e portas que se abriam somente após a digitação de uma senha no computador. Os dois subprogramas eram executados simultaneamente por processamento paralelo.

Os três pavimentos da casa eram servidos por um elevador, programado a fim de parar em cada andar. Para um funcionamento apropriado, o computador precisava saber em que andar estava o elevador. Este, construído com um brinquedo mecânico de armar, era acionado por um motor de corrente contínua e três chaves de contato, que serviam para detectar se o elevador estava em determinado andar. As crianças o programaram para descer ao térreo quando se pressionasse um botão; um outro botão enviava-o para o segundo andar; e um último, ao terceiro. As crianças se perguntaram: "O que acontecerá se um botão for acionado quando o elevador estiver em movimento?" — sua primeira resposta foi programar o elevador para ignorar todos os sinais até a parada. Mas, pensando melhor, decidiram que uma solução mais adequada seria o computador armazenar as instruções e implementá-las em seqüência.

Soluções simples

As características do C-LOGO permitiram desenvolver, de forma simples, um programa que sem ele seria bem mais complexo. Foram definidos como componentes necessários um detector de sinais, um seqüenciador e um controlador de elevador; o detector de sinais e o controlador do elevador agiam como processos paralelos. Os sinais eram detectados com FOREVER e transferidos para o seqüenciador. O controlador do elevador enviava uma mensagem ao seqüenciador, solicitando um destino: movia-o para o andar correto e solicitava novo destino. Após armazenar as instruções na ordem correta, o seqüenciador as enviava ao controlador. Empolgadas com o fato de o elevador estar na sua frente, assim como os meios de controlá-lo, as crianças persistiram até a resolução do problema.

Muitos educadores acreditam que um dos benefícios decorrentes da programação de robôs por crianças está no conceito de erro, que muda radicalmente. A experiência em tentativa e erro — e o conseqüente desenvolvimento de um método mais seqüencial de planejamento — traz o certo ou errado para um âmbito mais realístico, com correções constantes e mudanças para adaptação a novas situações.

Brincando com a tartaruga

O papel da tartaruga na sala de aula não se limita à geometria, trigonometria e programação. Já se construíram labirintos como parte de um projeto sobre mitologia (Teseu e o Minotauro) e sobre arquitetura de igrejas (o labirinto no chão da catedral de Reims). Um ambicioso projeto envolvia a construção, nas aulas de trabalhos manuais, de uma cidade em miniatura, que a tartaruga deveria percorrer.

Outros jogos com tartarugas envolviam pistas de corrida, em que cada criança programava a tartaruga, competindo com outras crianças. No jogo do Carteiro, sentadas em círculo, elas enviam mensagens umas para as outras, levadas pela tartaruga.

Dois jogos desenvolvidos especialmente para ajudar a ensinar o deslocamento angular e a distância são o Gira-tartaruga e o Empurra-tartaruga. No primeiro, posiciona-se a tartaruga no centro de um círculo dividido em seções; cada seção é marcada com um número diferente (equivalente aos pontos) e a criança precisa programar a tartaruga para girar e apontar para uma das seções; vence quem conseguir a contagem mais alta com o menor número de comandos no programa. O Empurra-tartaruga utiliza um método semelhante, mas com traçados retos.

Trajetória programada
O Big Trak é um brinquedo que pode ser programado de modo similar ao LOGO, utilizando o seu teclado numérico. Os comandos são constituídos de uma direção (direita, esquerda etc.) seguida de um número (até 99). O Big Trak pode armazenar e executar seqüencialmente até oito diferentes comandos.

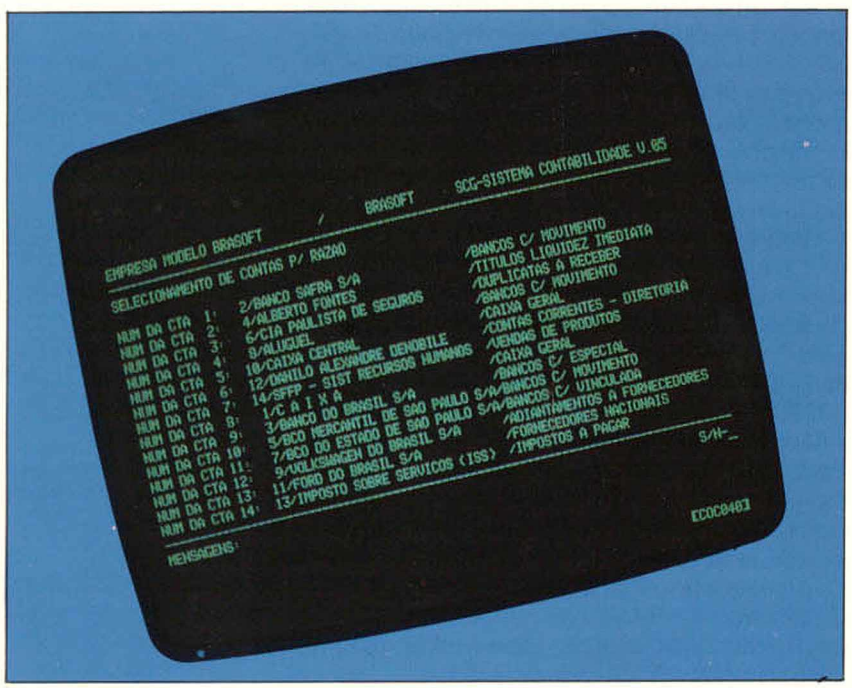
PACOTES FINANCEIROS

Um empresário poupará tempo e dinheiro ao automatizar sua indústria, se usar, em lugar de uma equipe especializada para desenvolver programas, um pacote de software já pronto.

A primeira idéia que ocorre a um pequeno ou médio empresário que decide automatizar sua empresa é comprar um microcomputador e contratar um programador para desenvolver seus programas. Poucos aprenderam que esse não é o melhor caminho.

Antes de mais nada, o empresário deve se conscientizar dos problemas existentes em sua indústria e conhecer suas reais necessidades de automação. Quando não for possível decidir sozinho, deverá recorrer a uma empresa especializada, que fará uma análise e traçará as diretrizes a seguir.

Diferente do que o usuário em geral pensa, o microcomputador não é a parte mais importante na automação comercial. Antes de pensar no equipamento, o empresário deve se preocupar com os softwares que mais correspondam às suas necessidades. E, para obtê-los, pode seguir dois caminhos: desenvolver seus próprios programas (através de um programador ou uma equipe especializada) ou comprar pacotes (programas prontos, disponíveis no mercado).



Pacotes de programas

Todo empresário, ao iniciar a automação de sua indústria, deve começar pelo departamento cujos problemas requerem soluções mais urgentes. Geralmente a necessidade de decisão é mais imediata no setor contábil.

Como a rotina da área contábil e financeira é muito parecida na maioria das empresas, algumas software houses desenvolvem pacotes de programas para facilitar o contato direto entre computadores e usuário. São de fácil manipulação e não necessitam de operadores especializados. O próprio empresário e seus funcionários podem utilizá-los sem problemas.

Existem várias sotware houses que, além de fornecerem os pacotes, também assessoram as empresas que querem automatizar seus departamentos. Uma delas é a Tecnosoft Informática, que atua há muitos anos no mercado prestando serviços a grandes indústrias. Na área financeiro-contábil, a Tecnosoft desenvolve três tipos de pacotes destinados a equipamentos de portes diferentes: o bureau de serviços destinado aos minicomputadores IBM; o pacote para micros, compatível com as linhas Cobra 210 e 305; e o pacote destinado aos mínis Cobra 480 e 500, que são equipamentos mais sofisticados.

O último dos três é operado com terminal on line, pelo próprio usuário. Não sofre interferência do centro de processamento de dados e dispensa o custo operacional. Trata-se de um sistema orçamentário com previsão antecipada de doze meses; emite relatórios mensais confrontando os valores orçados com os valores reais.

Outra característica desse sistema é o gerenciamento de todo o departamento, emitindo, a qualquer momento, a posição de cada conta, com os lançamentos atualizados, apresentando assim um fechamento real ou fictício.

O pacote compatível com a linha Cobra 210 é um pouco mais simples de processar e também pode ser operado pelo usuário através de menus de opções, que facilitam o trabalho para principiantes. Este, assim como o pacote para o Cobra 500, emite diário-razão, balancetes sintético e analítico, bem como catálogos de contas e estoques. Por suas características simplificadas, dirige-se mais para pequenas e médias empresas.

Diferentemente do pacote para a linha 210, o bureau de serviços visa mais a grande empresa. Sua abrangência equivale à dos demais sistemas.

Nos três sistemas, os totais são impressos de cima para baixo, permitindo melhor visualização das tabelas e apuração dos percentuais de participação em relação a cada conta de primeiro nível, isto é, a relação entre ativo, passivo, receita e despesa.

Outra empresa especializada em pacotes de sistemas é a MCS - Micromática Computadores e Sistemas, que também comercializa equipamentos. O mais recente lançamento da empresa, o Placon (Planejamento, Programação e Controle de Obras) destina-se à área de construção ci-

vil, mas também se adapta perfeitamente a outros ramos de atividades.

O sistema do Placon voltado ao setor financeiro identifica os custos de uma obra, faz uma projeção dos índices que vão atuar sobre cada um dos insumos e avalia, mensalmente, quais os gastos referentes à construção. Além disso, faz um cronograma econômico para prever o momento certo de utilização dos recursos. Com ele é possível calcular, ainda, fluxo de caixa, contas a pagar e receber, saldos bancários e toda a parte contábil.

Outro sistema de bastante aceitação nas empresas de médio e grande porte é o Portfólio, utilizado no planejamento e controle de investimentos em ações e mercado a termo. Permite ao analista financeiro decidir com segurança quanto ao melhor momento de comprar ações, vendê-las ou fazer aplicações em commodities e em mercadorias.

Um módulo do sistema Portfólio faz análise econômico-financeira de empresas. Dessa forma, uma companhia interessada em aplicações na área financeira pode ter um perfil das empresas de seu interesse, antes de decidir pela compra de papéis de uma delas. Logicamente não será apenas esse perfil que pesará na hora da decisão. O feeling comercial do empresário também é parte importante no processo, mas os dados fornecidos pelo software darão enorme respaldo à escolha. Todos os sistemas são desenvolvidos pela MCS para equipamentos de 8 e 16 bits.

Vantagens dos pacotes

Desenvolver programas não é privilégio de firmas especializadas. Qualquer empresário poderá ter seus próprios softwares, produzidos por um programador ou por uma equipe contratada para esse fim. Mas, ainda assim, estará correndo sérios riscos. Além de estar desembolsando uma soma — às vezes acima de seu orçamento —, existe a possibilidade de ficar insatisfeito com os resultados finais.

Podemos atribuir esses detalhes à relativa facilidade em se desenvolver um programa. Muitos programadores principiantes, dispostos a ganhar algum dinheiro extra, propõem-se a desenvolver programas para empresários com pouco ou nenhum conhecimento de informática. Já outros empresários, julgando ter bastante conhecimento da área, arriscam-se a desenvolver seus próprios programas, sem ajuda de profissionais. Os resultados, nesse caso, podem ser ainda mais desastrosos e os prejuízos decorrentes, incalculáveis.

Para um empresário com pouco ou nenhum conhecimento de informática, ou mesmo alguém com algum esclarecimento, mas cauteloso na hora de investir seu dinheiro, o indicado são os pacotes de sistemas. Embora tenham forma quase generalizada, os pacotes mostram-se flexíveis e adaptáveis à rotina de trabalho da maioria das empresas. As próprias software houses se encarregam de adaptar os sistemas às particularidades das empresas. Em geral, essas adaptações são feitas nos gráficos e relatórios, que raramente coincidem em duas ou mais empresas.

Todas essas particularidades têm outra função que o usuário nem sempre percebe: proteger as informações da empresa. Por serem quase padronizados, os pacotes apresentam características que os diferenciam entre si, para dificultar o acesso de pessoas estranhas aos dados da empresa. A Tecnosoft, por exemplo, fornece ao usuário uma senha secreta de acesso, a fim de proteger seus programas e informações.

De fácil entendimento, o pacote dispensa o uso de pessoal especializado e tem seu preço reduzido. Além disso, apresenta a vantagem de ser incorporado sem dificuldade à rotina de trabalho da empresa. Pode-se implantar o Placon, da MCS, por exemplo, numa empresa não automatizada e que necessite de um sistema complexo, em trinta dias. E os usuários levarão de seis a oito meses para dominar o processamento, o que é um período relativamente curto.

Como implantar o sistema

Existem vários estágios de conhecimento da área de informática. Na maioria das vezes, o empresário menos esclarecido se vê ludibriado por vendedores interessados apenas nas comissões, ou por pessoas de má fé, com o intento de prejudicar o funcionamento de sua empresa. Por isso, o industrial interessado em automatizar sua empresa deve procurar informações e esclarecer dúvidas junto a empresas idôneas especializadas em assessoria nessa área.

Uma das informações prioritárias que o usuário deve procurar obter antes de se lançar definitivamente no campo da informática é quanto à utilidade do computador e como ele pode ser adaptado às necessidades da empresa. Muitos executivos, ao pensar na automação de suas empresas, procuram cursos de computação para aprender as diversas linguagens fugindo de seus propósitos iniciais — o custo e os benefícios do equipamento. Um erro comum é pensar que o aprendizado do BASIC elementar bastará para desenvolver programas complexos. Estes exigem sofisticados recursos de programação, muitas vezes incorporando rotinas em ASSEMBLY.

Uma empresa de consultoria, além de orientar o empresário, fará um diagnóstico da empresa e estabelecerá suas prioridades em relação ao sistema. Isso porque estas podem não recair na área financeira. Dependendo do ramo de atividade da empresa, outros departamentos poderão ter mais urgência de automação.

Essa análise pode ser um simples diagnóstico ou então um plano diretor, diferenciando-se apenas pela abordagem dos detalhes. Em ambos os casos, os resultados são satisfatórios. Para uma grande empresa, com sistemas de automação mais complexos, vale a pena um plano diretor; já na pequena e média empresa, um diagnóstico é suficiente, pois, além de menor custo, usa linguagem acessível aos principiantes.

EXPERT

O Expert apresenta design moderno, amplos recursos e um conjunto de periféricos próprios. Produto da empresa brasileira Gradiente, segue o padrão MSX, de grande sucesso no mercado internacional.

Num esforço conjunto para padronizar o hardware e o software dos micros de 8 bits, mais de vinte fabricantes japoneses e europeus adotaram o padrão MSX (Microsoft Extended), que determina a configuração básica do micro, embora permita variações nos detalhes. No Brasil, os primeiros micros nesse padrão surgiram em 1985. A Gradiente, que ocupa privilegiada posição como fabricante brasileira de equipamentos de som, lançou nesse ano o Expert, com índice de nacionalização em torno de 80%.

Sua CPU é um processador Z80A de 8 bits, que opera em conjunto com dois processadores auxiliares, dedicados a vídeo e áudio. Essa arquitetura — considerada a mais avançada na área de 8 bits —, aliada a uma elevada freqüência de clock (3,58 MHz), proporciona ao Expert alta velocidade de processamento. Sua RAM, de 80 Kbytes, destina 16 ao processamento de vídeo e 64 ao usuário. A ROM, de 32 Kbytes, contém o sistema operacional e o interpretador MSX-BASIC, versão mundialmente aceita dessa linguagem e virtualmente idêntica à utilizada no IBM PC e compatíveis. Ambas as memórias, de leitura e de acesso aleatório, permitem expansão para até 256 Kbytes, por meio de cartuchos.

O teclado do Expert separa-se da unidade principal, para maior conforto na utilização. Há 89 teclas, incluindo o conjunto completo dos caracteres da língua portuguesa e um bloco numérico separado, tipo calculadora. Acima e à esquerda, encontram-se cinco teclas de função, que, em conjunto com a tecla [Shift], produzem até dez diferentes funções. À direita destas, há cinco teclas especiais, de grande utilidade na programação: [Stop], [Home], [Select], [Insert] e [Delete]. Ainda seguindo o padrão MSX, as quatro teclas de movimentação do cursor são de tamanho grande e dispõem-se num conjunto separado, para fácil operação.

A Gradiente desenvolveu para o Expert toda uma família de periféricos, harmonizados com seu moderno design e que otimizam sua utilização. O monitor de vídeo monocromático, de formato quadrado, tem controles externos de contraste, brilho, fundo invertido e redução de imagem. Apresenta 40 colunas de 24 linhas e resolução de 256 x 192 pixels. A Gradiente oferece também um adaptador para tevê comum; a saída RGB, para monitor colorido, encontra-se incorporada ao Expert.

Para o armazenamento de dados e programas, a Gradiente desenvolveu o Datacorder, um minicassete especial que lê e grava dados com rapidez e segurança. O Expert, porém, pode utili-

EXPERT

MICROPROCESSADOR

Z80A.

CLOCK

3,58 MHz.

MEMÓRIA

80 Kbytes de RAM e 32 Kbytes de ROM.

VÍDEO

Monitor monocromático.
Modo texto: 24 linhas de 40 caracteres.
Modo gráfico: resolução de 256 x 192 pixels.

TECLADO

Tipo QWERTY, com 89 teclas, com caracteres da língua portuguesa e um bloco numérico separado.

LINGUAGENS

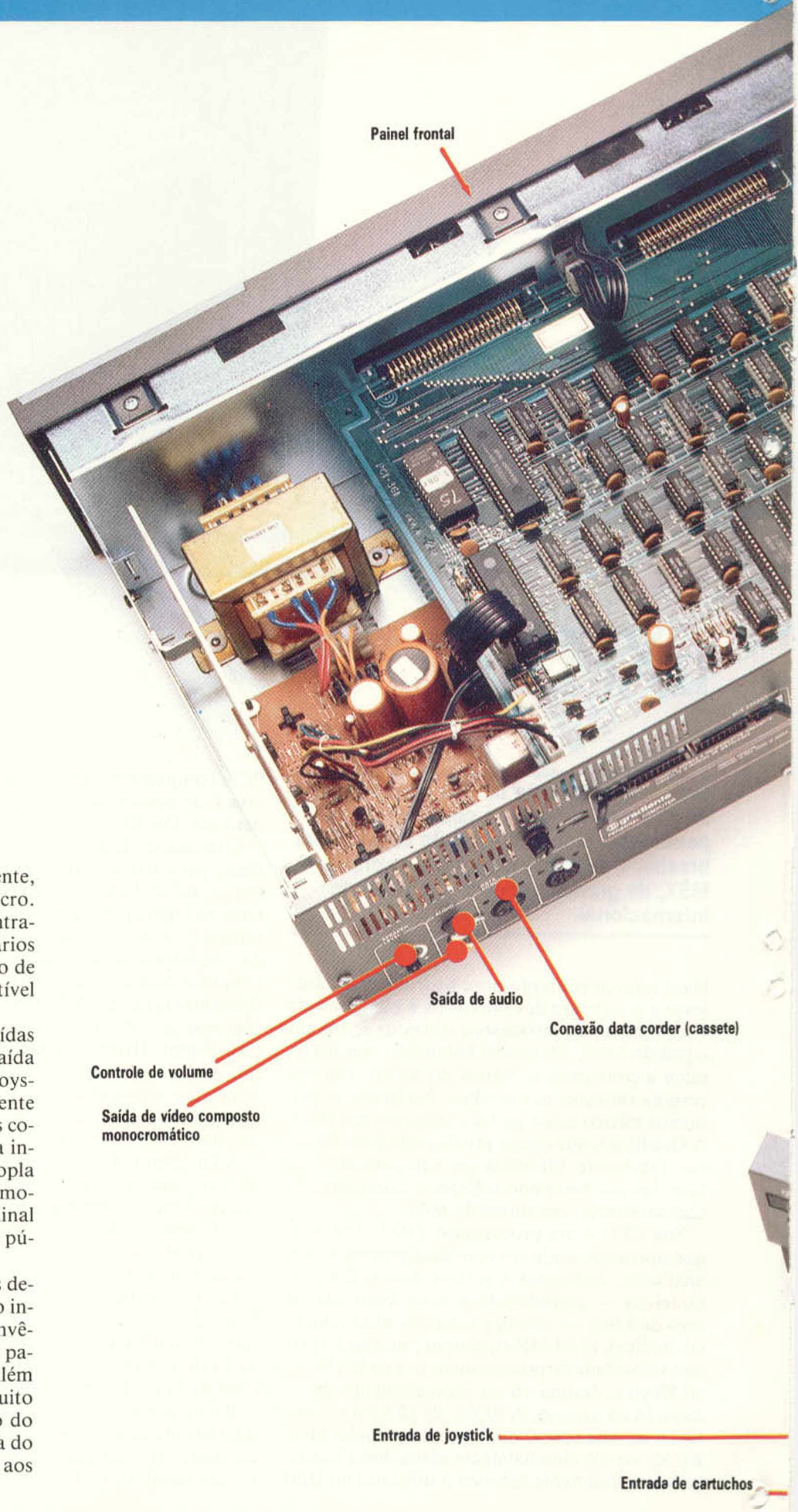
MSX-BASIC.

PERIFÉRICOS

Saída RGB; impressora padrão Centronics; Datacorder; gravadores cassete; joysticks e interface opcional para comunicações.

DOCUMENTAÇÃO

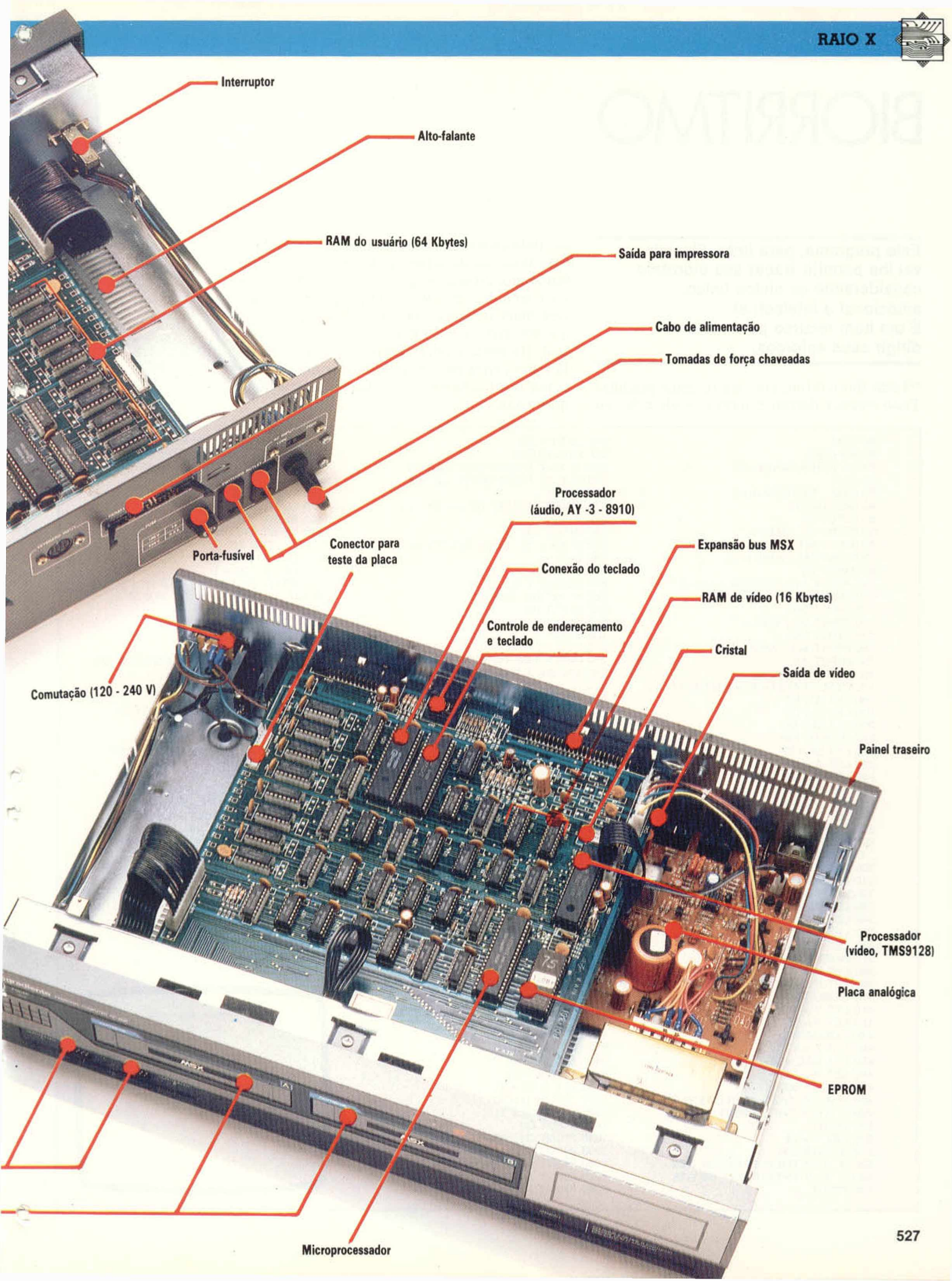
Dois manuais — um a respeito do equipamento e outro do MSX-BASIC — mais uma cartela contendo o resumo de operação do Expert.



zar qualquer gravador cassete; e, inversamente, o Datacorder é compatível com qualquer micro. Para programas aplicativos, existem duas entradas de cartucho na unidade central. Os usuários de disquetes deverão aguardar o lançamento de uma unidade de discos Gradiente compatível com o padrão MSX.

Na traseira da unidade central ficam as saídas para acoplamento de periféricos. Além da saída RGB, há uma para impressora e duas para joystick; o Expert aceita o de fabricação Gradiente ou qualquer outro padrão Atari. Além dessas conexões embutidas, a Gradiente oferece uma interface opcional para comunicações que acopla até oito diferentes expansões, inclusive um modem. Com isso, o Expert torna-se um terminal de videotexto com acesso a bancos de dados públicos, como o Cirandão, da Embratel.

Já existe grande quantidade de programas desenvolvidos para o padrão MSX no mercado internacional. A Gradiente tem, também, convênios com diversas software houses brasileiras para lançamento de programas aplicativos. Além dos utilitários básicos, a empresa investe muito no setor educacional. No Brasil, a exemplo do que ocorre no Japão e na Europa, a "cultura do MSX" vem abrindo novas possibilidades aos usuários de micros de 8 bits.

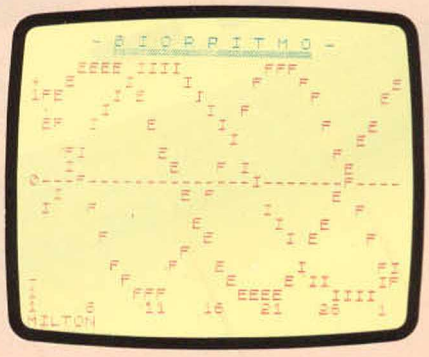

Interruptor
Alto-falante
RAM do usuário (64 Kbytes)
Saída para impressora
Cabo de alimentação
Tomadas de força chaveadas
Porta-fusível
Conector para teste da placa
Processador (áudio, AY -3 - 8910)
Expansão bus MSX
Conexão do teclado
RAM de vídeo (16 Kbytes)
Controle de endereçamento e teclado
Cristal
Saída de vídeo
Comutação (120 - 240 V)
Painel traseiro
Processador (vídeo, TMS9128)
Placa analógica
EPROM
Microprocessador

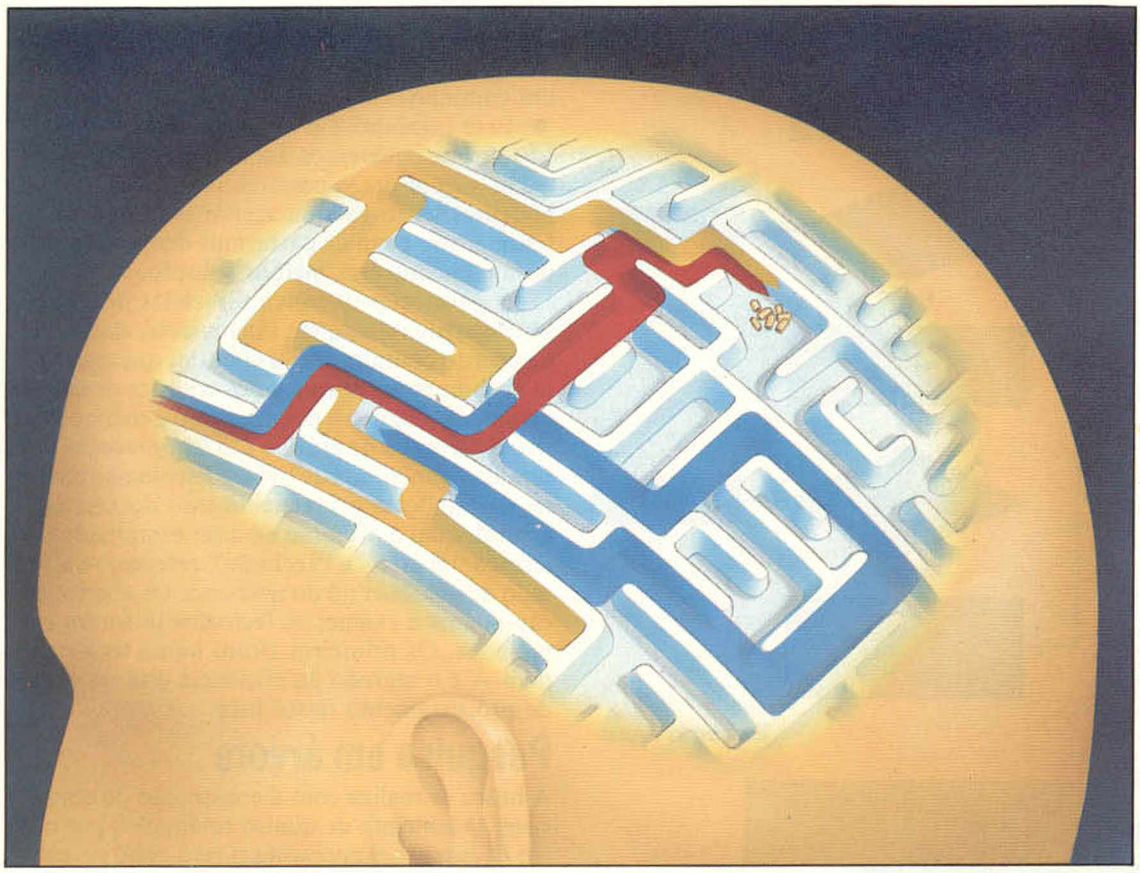
BIORRITMO

Este programa, para linha Sinclair, vai lhe permitir traçar seu biorritmo considerando os ciclos físico, emocional e intelectual. É um bom recurso para melhor dirigir seus esforços.

"Tudo flui e reflui, com movimento pendular. Tudo cresce e decresce; tudo ascende e descende; tudo caminha e volta ao ponto de partida". Essa frase, do dr. Krumm-Heller, é a síntese do biorritmo; ou seja, o ritmo que rege a fisiologia e a mente dos seres vivos. O programa a seguir determina as fases do biorritmo humano conforme seus três ciclos: físico, emocional e intelectual. Há ainda a opção gráfica, que traça na tela o biorritmo para os próximos trinta dias e o copia na impressora. Para reiniciar, aperte qualquer tecla.

```
10 REM
BIORRITMO
PARA LINHA SINCLAIR
20 CLS
30 REM  PEDE DADOS
40 GOSUB 710
50 INPUT D1
60 PRINT D1,,"MES = ";
70 INPUT M1
80 PRINT M1,,"ANO = ";
90 INPUT A1
100 PRINT A1,,,,"ENTRE A DATA D
E NASCIMENTO:","DIA = ";
110 INPUT D0
120 PRINT D0,,"MES = ";
130 INPUT M0
140 PRINT M0,,"ANO = ";
150 INPUT A0
160 PRINT A0,,,,
170 REM  CALCULA N.º DE DIAS
180 LET A = A0
190 LET D = D0
200 LET M = M0
210 GOSUB 740
220 LET DI = DF
230 LET A = A1
240 LET D = D1
250 LET M = M1
260 GOSUB 740
270 LET DEL = DF - DI
280 REM  IMPRIME NA TELA
290 PRINT "DIAS VIVIDOS = ,DEL,,
,"CICLOS = "
300 PRINT
310 LET A$ = "FISICO:"
320 LET B$ = "EMOCIONAL:
330 LET C$ = "INTELECTUAL:"
340 LET D$ = "- - > +"
350 LET E$ = "- - > -"
360 LET P = 23
370 GOSUB 800
380 LET W = X
390 LET P = 28
400 GOSUB 800
410 LET Y = X
420 LET P = 33
430 GOSUB 800
440 LET Z = X
450 LET DEL = DEL + 1
460 LET P = 23
470 GOSUB 800
480 IF X > W THEN PRINT A$;W;D$
490 IF X < W THEN PRINT A$;W;E$
500 PRINT
510 LET P = 28
520 GOSUB 800
530 IF X > Y THEN PRINT B$;Y;D$
540 IF X < Y THEN PRINT B$;Y;E$
550 PRINT
560 LET P = 33
570 GOSUB 800
580 IF X > Z THEN PRINT C$;Z;D$
590 IF X < Z THEN PRINT C$;Z;E$
600 PRINT
610 PRINT AT 21,6;"GRAFICO ? (S
/N)"
620 INPUT K$
630 IF K$ = "S" THEN GOTO 840
640 CLS
650 STOP
660 GOSUB 710
670 PRINT D1;"/";M1;"/";
680 GOTO 100
690 CLS
700 STOP
710 CLS
720 PRINT TAB 11;"BIORRITMO";AT
2,0;"ENTRE A DATA DESEJADA:","D
IA = ";
730 RETURN
740 IF M > = 3 THEN GOTO 770
750 LET A = A + 1
760 LET M = M + 12
770 LET M = M + 1
780 LET DF = INT (A*365.25) + INT (
M*30.6) + D
790 RETURN
800 LET X = DEL/P
810 LET DF = X - INT X
820 LET X = INT (SIN (2*DF*PI)*10
0)/100
830 RETURN
840 REM  GRAFICO
850 CLS
860 PRINT "QUAL O SEU NOME ?"
870 INPUT N$
880 PRINT N$
890 PRINT "DE A PRIMEIRA DATA:"
,"DIA = ";
900 DIM S(12)
910 LET S(1) = 31
920 LET S(2) = 28
930 LET S(3) = 31
940 LET S(4) = 30
950 LET S(5) = 31
960 LET S(6) = 30
970 LET S(7) = 31
980 LET S(8) = 31
990 LET S(9) = 30
1000 LET S(10) = 31
1010 LET S(11) = 30
1020 LET S(12) = 31
1030 INPUT D1
1040 PRINT D1,,"MES = ";
1050 INPUT M1
1060 PRINT M1,,"ANO = ";
1070 INPUT A1
1080 PRINT A1
1090 LET DEL = 365.25*A1 + 30.6*M1 + D
1 - 365.25*A0 - 30.6*MØ - DØ
1100 LET S(2) = S(2) + (A1/4 = INT (A1
/4))
1110 PAUSE 120
1120 CLS
1130 SLOW
1140 PRINT TAB 5;" - B I O R R I
T M O -";TAB 7;"
1150 LET D1 = D1-2
1160 LET E = D1 + 1
1170 PRINT AT 5,0;" + ";TAB 0;"1";
AT 19,0;" - ";TAB 0;"1";AT 12,0;"0
----------"
1180 FOR G = D1 TO D1 + 30
1190 LET DEL = DEL + 1
1200 LET E = E + 1
1210 IF E = S(M1) + 1 THEN GOSUB 139
0
1220 LET P = 23
1230 GOSUB 800
1240 PRINT AT 12-8*X,1 + G - D1;"F"
1250 LET P = 28
1260 GOSUB 800
1270 PRINT AT 12-8*X,1 + G - D1;"E"
1280 LET P = 33
1290 GOSUB 800
1300 PRINT AT 12-8*X,1 + G - D1;"I"
1310 IF (G-D1)/5 = INT ((G - D1)/5)
THEN PRINT AT 21,G - D1;E
1320 NEXT G
1330 REM  COPIA NA IMPRESSORA
1340 COPY
1350 LPRINT N$
1360 PAUSE 4E4
1370 FAST
1380 RUN
1390 LET E = 1
1400 LET M = M + 1
1410 IF M = 13 THEN LET M = 1
1420 RETURN
```





O caminho
O diagrama mostra três caminhos possíveis através do labirinto, percorridos graças a três diferentes estratégias de busca: aleatória, heurística e sistemática. Neste exemplo, o método heurístico atinge o objetivo mais rapidamente que os outros, embora seja possível projetar um labirinto em que isso não acontece. Na prática, os problemas envolvendo labirintos costumam ser solucionados por uma combinação dos métodos heurístico e exaustivo (aquele em que todos os pontos são considerados).

Heurístico

Sistemático

Aleatório

BUSCA INTELIGENTE

Encontrar o caminho num labirinto é um problema clássico da inteligência artificial, exigindo avançadas técnicas de busca. Mostramos aqui três dessas técnicas, incorporadas num único programa computacional.

Imagine três ratos num labirinto, onde existe um pedaço de queijo. Um deles cambaleia durante alguns minutos e então adormece (o insensível experimentador colocou gim em sua água). O segundo rato é mais metódico: apóia contra a parede a pata esquerda dianteira e prossegue, virando à esquerda a cada bifurcação e retrocedendo quando encontra um beco sem saída. Termina por atingir o objetivo — mas, a essa altura, o terceiro rato já comeu o alimento. Este, de olfato mais aguçado, periodicamente farejava o ar e avançava pelo caminho que pressentia estar conduzindo-o para mais perto do delicioso cheiro. Sua estratégia de busca revelou-se, sem dúvida, a mais inteligente das três.

A busca é um conceito-chave no campo da inteligência artificial. Seu filho pode estar no quintal brincando de caça ao tesouro enquanto você, comandante sentado em sua poltrona, faz palavras cruzadas: tanto você quanto ele estão procurando alguma coisa. O exemplo dos ratos no labirinto mostra-se proveitoso, visto que muitos problemas diferentes podem ser abordados por meio de técnicas de busca. De fato, os três ratos ilustram diferentes tipos:

- Busca aleatória (o "passeio do bêbado");
- Busca exaustiva (enumeração sistemática);
- Busca heurística (exploração dirigida).

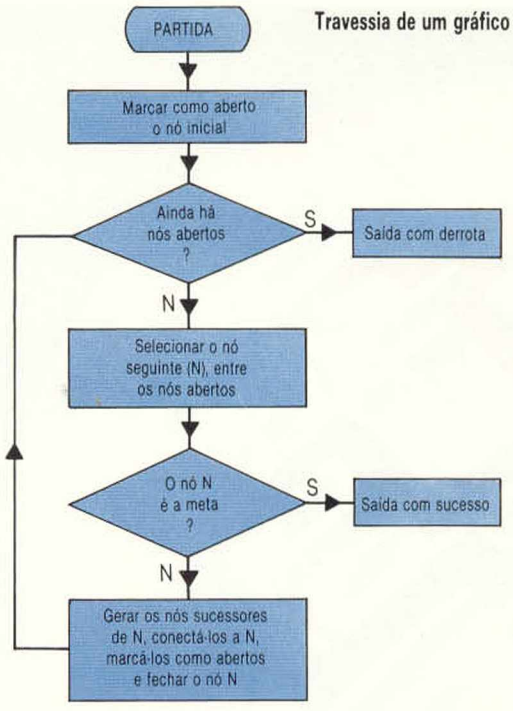
A terceira estratégia é considerada uma abordagem mais inteligente do problema porque, em geral, exige menos esforço para atingir a solução. Todos os métodos heurísticos, porém, devem contar com algum meio de saber quando a busca se aproxima do objetivo (por exemplo, o sentido do olfato no rato). Sem esse conhecimento, não se pode agir de modo inteligente: será preciso confiar na enumeração sistemática das possibilidades, ou esgotar todas elas.

Achar o caminho num labirinto é um típico exercício de busca. (Muitos problemas não espaciais, entretanto, tais como a integração de uma expressão matemática simbólica, podem ser abordados com o mesmo referencial de busca.)

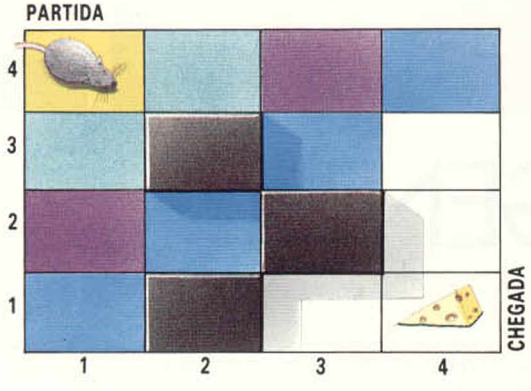
Busca ramificada

A maioria dos métodos de busca empregados pela inteligência artificial envolve a geração de uma árvore, criada a partir do esquema do labirinto a ser atravessado. Na árvore aqui mostrada, toma-se o ponto de partida como raiz e ramificam-se os níveis seguintes mediante a consideração de todos os movimentos possíveis que partem do nó inicial. Movimentos em diagonal não são permitidos. O fluxograma aponta o algoritmo básico para a geração da estrutura da árvore.

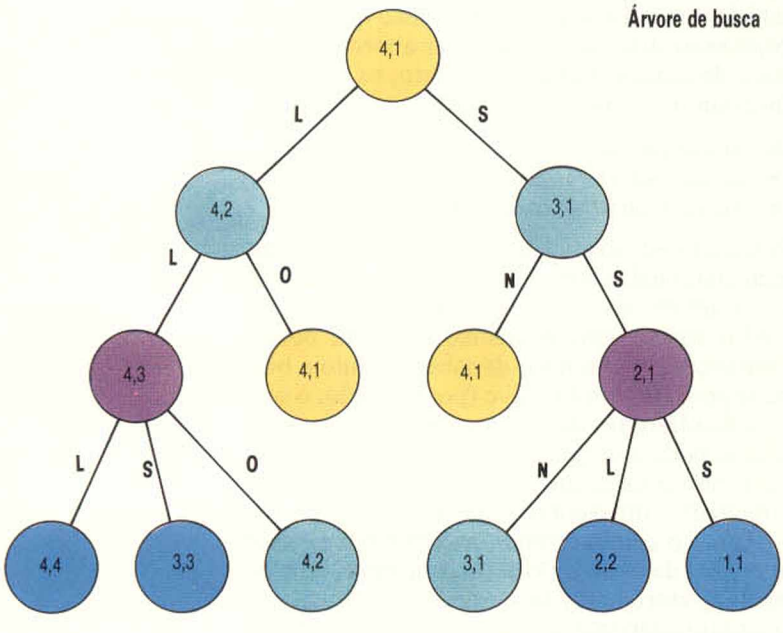
Travessia de um gráfico



Um labirinto simples



Árvore de busca



Existem dois meios para se julgar a qualidade de um método de busca:

- Qual o tempo necessário para se encontrar um caminho?
- Esse caminho é o mais econômico?

Idealmente, queremos encontrar a solução ótima no menor prazo possível. Na prática, porém, podemos ser forçados a aceitar uma solução subótima, sob pena de gastarmos demasiado tempo até encontrar a melhor solução.

As estratégias de busca utilizadas pela inteligência artificial são, em sua maioria, elaboradas de acordo com um plano comum, que possibilita atribuir pesos distintos aos critérios de desempenho. O diagrama de algoritmo genérico relativo a travessia de gráfico *(ao lado)* mostra uma família de métodos de busca. Escolhe-se um deles definindo-se o preenchimento do retângulo central: qual o próximo nó a ser examinado? Os termos "aberto" e "fechado" referem-se ao estado de qualquer nó do percurso. Os abertos são candidatos a exame; os fechados já foram examinados. Os primeiros ficam numa lista de espera — e o segredo da eficiência está na ordem de processamento dessa lista.

Pesquisa em árvore

A busca se realiza com a construção de uma árvore. O labirinto de quatro retângulos por quatro *(à esquerda)* apresenta o rato-robô no canto superior esquerdo e o queijo no inferior direito; de cada retângulo, o rato pode ir para norte, sul, leste ou oeste, embora alguns (os escuros) estejam bloqueados. Se traçarmos as opções existentes no início do percurso, veremos uma estrutura de busca em árvore.

A partir do retângulo inicial (4,1), o rato pode mover-se para leste (4,2) ou para sul (3,1). Saindo de (4,2), pode ir para leste ou para oeste; e, de (3,1), vai para norte ou para sul. Caso se mova para leste e depois para oeste (ou para o sul e depois para o norte), voltará ao ponto de partida, o que não será uma decisão muito inteligente. Isso mostra, porém, que o mesmo retângulo pode aparecer em diferentes nós da árvore, correspondentes aos vários caminhos para se chegar até eles.

A "busca exaustiva", um método sistemático de explorar a árvore, investiga os nós em ordem de proximidade ao ponto de partida, ou raiz da árvore. Assim, considera cada seqüência de N movimentos (no nível N da árvore) antes de qualquer seqüência de N+1 movimentos. Objetiva encontrar um caminho que exija o menor número de passos; como todos os passos são igualmente dispendiosos, o caminho será uma solução de dispêndio mínimo — que, entretanto, pode demorar a ser encontrada. À medida que se multiplicam as ramificações da árvore, o tempo necessário para se atingir o objetivo aumenta exponencialmente.

Aperfeiçoar uma busca exaustiva requer uma fonte de informações heurísticas relativas à distância do objetivo. Tomemos como exemplo um

bairro onde os quarteirões sejam quadrados exatos. Para irmos de A até B devemos andar um certo número de quarteirões para norte ou sul, e outros tantos para leste ou oeste. Da mesma forma, no labirinto, um rato inteligente pode calcular a quantos retângulos está do alvo; sabendo quando a busca se aproxima do final, ele pode andar mais depressa.

O rato do início deste artigo seguia uma regra simples: chegar cada vez mais perto da meta. É a estratégia "subir o morro", assim chamada por ser comparável à escalada de uma montanha envolta na névoa: basta mover-se sempre para cima. Pode ser bem mais rápida do que a busca exaustiva, mas não garante que se encontre o caminho ótimo.

O método chamado A-asterisco (A*) consiste num compromisso meio-termo entre "subir o morro" e a busca exaustiva. O nó a ser examinado é selecionado com base na soma HD + DP, onde HD representa a estimativa heurística da distância e DP a distância percorrida. Quanto mais correta a estimativa de HD, mais eficaz a busca. Se houver erro em HD, convém que seja por subestimar a distância e não por superestimá-la.

Neste programa-exemplo, utilizamos a estrutura de um só programa para implementar os três métodos, com pequenas alterações. As únicas diferenças são evidenciadas na escolha do nó examinado em seguida:

Busca exaustiva — escolha a menor DP;
Subir o morro — escolha a menor HD;
Algoritmo A* — escolha a menor HD + DP.

Programa de procura no labirinto

```
1000 REM ******************************
1010 REM ** RELATORIO 2.1            **
1020 REM ** PROGRAMA DE PROCURA NO   **
1030 REM ** LABIRINTO                **
1040 REM ******************************
1050 MH = 17:MW = 25: REM  ALTURA E PROFUNDIDADE DO LABIRINTO
1060 SI = 256: REM  LIMITES DA ARVORE
1070 WA = 1:RR = 2:FO = 3:BN = 4:BL = 5
1080 W1 = 1: REM  PESO PARA SF
1090 W2 = 2: REM  PESO PARA HD
1100 DIM M(MH + 1,MW + 1): REM  O LABIRINTO
1110 DIM C$(5): REM  CARACTERISTICAS DO LABIRINTO
1120 DIM P(SI),S(SI),N(SI),H(SI)
1130 REM  CAMINHO,PASSOS,NO,DISTANCIA HEURISTICA
1140 REM
1150 REM  --RATO NO LABIRINTO:
1160 GOSUB 1360: REM  FAZER O LABIRINTO
1170 NC = 0: REM  NUMERO DE NOS EXAMINADOS
1180 K = 0: REM  CONTADOR
1190 GOSUB 1660: REM  LIMPE TODOS OS CAMINHOS
1200 N(1) = 2 * MW + 2: REM  PRIMEIRO NO ABERTO
1210 S(1) = 0:H(1) = FR - 1 + (FC - 1)
1220 P(1) = 0: REM  SEM PREDECESSOR
1230 REM  --LOOP PRINCIPAL:
1240 REM  ****LOOP PRINCIPAL****
1250 GOSUB 1770: REM  ESCOLHE PROXIMO NO S
1260 NC = NC + 1
1270 VT( ) 22: PRINT NC,SR,SC,H(S);"          "
1280 GOSUB 1880: REM  GERA SUCESSOR
1290 IF FR <  > SR AND NC < 300 THEN 1240
1295 IF FC <  > SC AND NC < 300 THEN 1240
1300 VTAB 23: PRINT "PROCURA TERMINADA"
1310 IF FR = SR AND FC = SC THEN  GOSUB 2290: REM  RETRACE OS PASSOS
1320 IF NC > 300 THEN  PRINT "FALHOU"
1330 PRINT NC;"NOS EXAMINADOS."
1340 END
1350 :
1360 REM  ROTINA PARA CRIAR O LABIRINTO:
1370 FOR P = 1 TO MH + 1
1380 FOR R = 1 TO MW + 1
1390 IF  RND (1) < 0.28 THEN M(P,R) = WA
1395 IF  RND (1) > 0.28 THEN M(P,R) = BL
1400 IF P = 3 OR R = 3 THEN M(P,R) = BL
1410 IF P = 1 OR R = 1 THEN M(P,R) = WA
1420 IF P > MH OR R > MW THEN M(P,R) = WA
1430 NEXT R: NEXT P
1440 FR = 2 +  INT ( RND (1) * (MH - 1)): REM  FILEIRA DE COMIDA
1450 FC = 4 +  INT ( RND (1) * (MW - 3)): REM  COLUNA DE COMIDA
1460 M(FR,FC) = FO
1470 M(2,2) = RR: REM  RATO ROBO
1480 C$(BL) = " "
1490 C$(WA) =  CHR$ (88): REM  MANCHA
1500 C$(BN) = " "
1510 C$(RR) = "R"
1520 C$(FO) = "F"
1530 GOSUB 1560: REM  MOSTRE O
1540 RETURN
1550 :
1560 REM  --ROTINA PARA MOSTRAR O LABIRINTO
1570 HOME
1580 FOR R = 1 TO MH + 1
1590 FOR C = 1 TO MW + 1
1600 HTAB C: VTAB R: PRINT C$(M(R,C));
1610 NEXT C
1630 NEXT R
1640 RETURN
1650 :
1660 REM  --ROTINA PARA LIMPAR A ARVORE:
1670 DD = 9999: REM  MORTO
1680 FOR Q = 1 TO SI
1690 P(Q) = 0
1700 S(Q) = DD
1710 N(Q) = 0
1720 H(Q) = DD
1730 NEXT Q
1740 NN = 2: REM  PROXIMO NO LIVRE
1750 RETURN
1760 :
1770 REM  --ESCOLHE O MELHOR NO S
1780 S = 1:BN = DD
1790 FOR I = 1 TO SI
1800 V = S(I) + W1 +  ABS (H(I)) * W2
1810 IF V < BN AND H(I) <  > 0 THEN S = I:BN = V
1820 NEXT I
1830 IF S = 1 THEN 1835
1835 HTAB 0: VTAB 20: PRINT "EXPLORANDO...."
1840 SR =  INT (N(S) / MW)
1850 SC = N(S) - MW * SR
1860 RETURN
1870 :
1880 REM  --ROTINA PARA GERAR SUCESSORES:
1890 IF H(S) = 0 THEN  RETURN : REM  FEITO
1900 REM  --NORTE
1910 Y = SR - 1:X = SC
1920 IF Y > 1 THEN  GOSUB 2090
1930 REM  --ESTE
1940 Y = SR:X = SC + 1
1950 IF X <  = MW THEN  GOSUB 2090
1960 REM  --SUL
1970 Y = SR + 1:X = SC
1980 IF Y <  = MH THEN  GOSUB 2090
1990 REM  --OESTE
2000 Y = SR:X = SC - 1
2010 IF X > 1 THEN  GOSUB 2090
2020 REM  --FECHA TAMBEM O NO S
2030 H(S) =  - H(S)
2040 IF H(S) > 0 THEN  PRINT "NOSSA"
2050 HTAB SC: VTAB SR: PRINT " "
2060 M(SR,SC) = DN
2070 REM  CELULAS BRANCAS NA TELA
2080 :
2090 REM  ROTINA PARA ABRIR UM NO
2100 IF M(Y,X) = DN THEN  RETURN
2105 AW = WA
2110 IF M(Y,X) = AW THEN  RETURN
2120 REM  --PRIMEIRO ACHE A LOCALIZACAO LIVRE
2130 NX = 0
2140 REM  **LOOP PARA LOCALIZAR
2150 IF S(NN) <  > 0 THEN NX = NX + 1:NN = NN + 1
2160 IF NN > SI THEN NN = 1
2170 IF NX > SI THEN  PRINT "ESTA CHEIO!": STOP
2180 IF S(NN) <  > 0 THEN 2140
2190 REM  --AGORA ABRA O
2200 IY = I + Y * MW
2210 N(NN) = IY
2220 P(NN) = S
2230 S(NN) = S(S) + 1
2240 H(NN) =  ABS (Y - FR) +  ABS (X - FC)
2250 HTAB X: VTAB Y: PRINT "+"
2260 REM  O MOSTRA NA TELA
2270 RETURN
2280 :
2290 REM  ROTINA PARA CAMINHO DE VOLTA
2300 ST = S(S)
2310 FOR Q = 1 TO 10000: NEXT Q: GOSUB 1560
2320 HTAB FC: VTAB FR: PRINT C$(FO);
2330 REM  **IMPRIMA O CAMINHO**
2340 S = P(S): REM  NO PARENTE
2350 IY = N(S): REM  COORDENADAS
2360 Y =  INT (IY / MW)
2370 X = IY - Y * MW
2380 M(Y,X) = RR: REM  PEGADA DE RATO!
2385 IF X <  > 0 OR Y <  > 0 THEN 2390
2386 IF X = 0 OR Y = 0 THEN 2400
2390 HTAB X: VTAB Y: PRINT "*"
2400 IF S > 0 THEN 2330
2410 HTAB 2: VTAB 2: PRINT C$(RR)
2420 VTAB 20: PRINT "CAMINHO DE";ST;"PASSOS"
2430 PRINT NC;"NOS FECHADOS."
2440 RETURN
```

As linhas do labirinto

O labirinto está contido numa matriz bidirecional M (,), e as estruturas de dados para a árvore de busca, contidas nas matrizes P( ), S( ), N( ) e H( ). O programa combina os fatores custo-até-aqui e custo-daqui-até-o-objetivo, que podem ser ponderados alternando-se os valores W1 e W2 nas linhas 1080 e 1090. Com os valores da listagem, a estratégia vigente é do tipo "subir o morro"; o método heurístico utilizado baseia-se no exemplo dos quarteirões simétricos. Fazendo-se prevalecer o método heurístico (com W2 > W1), acelera-se a busca.

O programa foi escrito para os micros da linha Apple. A versão para o TK 2000 é igual à da Apple.
Na versão para o CP-500 e compatíveis TRS, trocar as seguintes linhas:

Apple e TK 2000:
Na listagem ao lado, eliminar a linha 1230 e substituir na linha 1520 'F' por 'Q' (queijo em vez de food)

CP-500 e TRS:
Trocar as seguintes linhas:

```
1050 MH = 9: MW = 15 ...
1270 PRINT Q768, NC ...
1300 PRINT Q832,
     "PROCURA ...
1390 IF RND(0)<0.28 THEN
     M(P,R) = WA ELSE
     M(P,R) = BL
1395      [eliminar]
1440 ...RND(0)...
1450 ...RND(0)...
1600 PRINT Q R*64 + C,
     C$(M(R,C));
1830 IF S = 1 THEN PRINT
     Q 704,
     "EXPLORANDO"
1835      [eliminar]
2050 PRINT Q SR*64 + SC,
     " ";
2250 PRINT Q Y*64 + X,
     "+";
2320 PRINT Q FR*64 + FC,
     C$(FO);
2390 PRINT Q Y*64 + X, "*";
2410 PRINT Q 130, C4(RR);
2420 PRINT Q 704,
     "CAMINHO DE"; ST;
     PASSOS
```

ROTINAS GENÉRICAS

Para tornar rotinas em ASSEMBLY relocáveis, de modo a serem executadas qualquer que seja sua localização na memória, utilizamos símbolos e rótulos no lugar de endereços e valores absolutos.

O ASSEMBLY é uma linguagem de programação de baixo nível, ou seja, compõe-se de comandos "primitivos", que a CPU processa um a um. Por isso, o programador vê-se obrigado a escrever e reescrever rotinas para realizar tarefas essenciais, que já se encontram incorporadas aos comandos de uma linguagem de alto nível — gerenciamento de entrada e saída, por exemplo, ou operações aritméticas com 2 bytes. A melhor solução é montar uma biblioteca — em disco, fita ou papel — com as rotinas mais usadas e anexá-las aos novos programas quando necessário.

Esse processo, contudo, envolve dois grandes problemas. O primeiro é a dificuldade de se escreverem rotinas importantes, e às vezes longas, genéricas o suficiente para funcionarem em diferentes programas sem adaptações. O segundo consiste em escrever rotinas que não residam permanentemente num único local da memória, mas possam ser reposicionadas, por meio de uma nova compilação, com um endereço ORG diferente, continuando ali a desempenhar as mesmas funções.

Ambos os problemas derivam da questão da generalidade e compatibilidade, também existente no BASIC, e as soluções também são as mesmas utilizadas pelos programadores dessa linguagem: usar variáveis para passar valores do programa principal para uma sub-rotina; usar nas sub-rotinas variáveis locais, de modo a torná-las independentes do programa principal; evitar o uso de valores absolutos, sejam numéricos ou alfabéticos, bem como números de linhas.

Já nos habituamos a considerar os endereços de memória em ASSEMBLY como equivalentes às variáveis em BASIC; isto é, os comandos atuam sobre o conteúdo dos endereços, da mesma forma que um programa em BASIC manipula o conteúdo das variáveis. Até agora, em geral, nos referimos às posições de memória pelo seu endereço absoluto — de início um hábito conveniente, mas que devemos abandonar em nome da generalidade. Em vez de endereços e valores absolutos, passaremos a usar símbolos; em vez de variáveis e número de linhas, usaremos uma série de pseudocódigos operacionais. Já vimos exemplos de ambos os tipos, como este programa:

6502			Z80		
DATA1	EQU	$12	DATA1	EQU	$12
DATA2	EQU	$79	DATA2	EQU	$79
	LDA	DATA1	LD	A,(DATA1)	
LOOP	ADC	DATA2	ADC	A,(DATA2)	
	BNE	LOOP	JR	NZ,LOOP	
	RTS		RET		

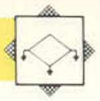
Temos aqui dois tipos de símbolos — dois valores e um rótulo — usados como operandos de instruções em ASSEMBLY. Esse recurso torna a rotina genérica e também relocável. As únicas quantidades absolutas são os valores de DATA1 e DATA2, que podem inicializar-se no programa principal, e não no início da própria rotina.

Há outros pseudocódigos à nossa disposição — DB ("definir byte"), DW ("definir palavra") e DS ("definir armazenamento"). (Note que, assim como ORG e EQU, esses nomes podem variar de um compilador para outro.) Os três comandos permitem inicializar e alocar posições de memória, como neste exemplo:

			ORG	$D3A0
D3A0	5F	LABL1	DB	$5F
D3A1	CE98	LABL2	DW	$98CE
D3B3		LABL3	DS	$10
D3B3		DATA1	EQU	LABL3

TABELA DE SÍMBOLOS
LABL1 = D3A0: LABL2 = D3A1: LABL3 = D3A3
DATA1 = D3A3
ASSEMBLY COMPLETO - NÃO HÁ ERRO

Nessa listagem completa em ASSEMBLY, produto do programa compilador, vemos, na parte inferior, uma tabela que associa os símbolos utilizados no programa aos valores que representam. Há várias coisas importantes a observar. A linha que começa com LABL1 utiliza o pseudocódigo DB. Vemos na listagem que o comando ORG atribuiu a LABL1 o endereço $D3A0, o que a tabela de símbolos confirma. O efeito de DB aqui é colocar o valor $5F no byte endereçado por LABL1; assim, a posição de memória $D3A0 inicia com o valor $5F, como vemos na coluna do código de máquina.



Note também que LABL2 representa o endereço $D3A1; contudo, como DW inicializa uma "palavra", isto é, 2 bytes consecutivos, o valor $98CE ficará armazenado nas posições $D3A1 e $D3A2 sob a forma baixo-alto (lo-hi), o que também se vê na coluna "Código de máquina". Como DW sempre converte seus operandos na forma baixo-alto, usa-se com freqüência esse pseudocódigo para inicializar posições indicadoras, isto é, que apontam para outros endereços. A posição LABL2, ou $D3A1, enquadra-se nessa categoria, pois aponta para o endereço $98CE.

Observe ainda que a instrução DS $10 acrescenta $10 ao contador do programa. Isso fica mais claro na tabela de símbolos que na própria listagem: LABL3 representa a posição $D3A3 (que se segue à instrução anterior), embora na listagem esse valor pareça ser $D3B3. Este último é o endereço da instrução que se segue a DS; assim, DS $10 reservou um bloco de 16 bytes (de $D3A3 a $D3B2 inclusive) entre as duas instruções. Esse processo assemelha-se à inserção de longas linhas REM num programa em BASIC a fim de criar um espaço não utilizado na área reservada ao texto do programa, para então processá-la com PEEKs e POKEs, como se fosse uma área de programa em código de máquina.

A última instrução do programa utiliza EQU, que dá a um símbolo o mesmo valor de outro; assim, DATA1 assume o valor $D3A3, atribuído a LABL3. Aqui pode haver uma confusão: LABL3 é a representação simbólica do endereço $D3A3; portanto, a instrução DATA1 EQU LABL3 significa "o símbolo DATA1 passa a assumir o mesmo significado e o mesmo valor do símbolo LABL3". O fato de a instrução DB ter colocado o conteúdo $5F na posição $D3A3 em nada altera o significado dos símbolos LABL3 e DATA1. Aqui nos deparamos com uma das maiores dificuldades do aprendizado básico do ASSEMBLY: separar claramente os conceitos de "endereço" e "conteúdo do endereço". Talvez, ao aprender BASIC, você tenha esbarrado na mesma dificuldade de separar as variáveis de seus conteúdos, que são os valores que elas assumem.

À primeira vista, o comando DB parece uma duplicata de EQU, mas essa impressão é falsa. LABL1 aqui significa "a posição $D3A0", e a instrução DB $5F atribuiu o valor $5F a esse byte; contudo, embora o valor de LABL1 já esteja estabelecido, o conteúdo da posição que ele representa pode alterar-se a qualquer momento do programa (por exemplo, armazenando ali o conteúdo do acumulador). Da mesma forma, DATA1 é um símbolo cujo valor estabeleceu-se pela instrução EQU; esse valor não se altera pela execução do programa. Também assim, LABL3 indica o início de uma área de dados de 16 bytes, cujo conteúdo se altera durante a execução do programa; o próprio LABL3, no entanto, é inalterável.

			ORG	$D3A0
D3A0	4D4553	LABL1	DB	'MESSAGE 1'
D3A9	CE98	LABL2	DW	$98CE
D3BB		LABL3	DS	$10
D3BB		DATA1	EQU	LABL3

TABELA DE SÍMBOLOS
LABL1 = D3A0: LABL2 = D3A9: LABL3 = D3AB
DATA1 = D3AB
ASSEMBLY COMPLETO - NÃO HÁ ERRO

Essas são algumas das possibilidades de ação dos pseudocódigos; há outras, ainda. Considere a seguinte nova versão da rotina anterior:
A instrução DB tem como operando o string 'MESSAGE1', e o compilador inicializou os endereços de $D3A0 a $D3A8 com os valores ASCII dos caracteres entre aspas. Isso transparece na coluna de endereços, e se confirma pela coluna "Código de máquina", onde se vê que o conteúdo dos 3 bytes de $D3A0 a $D3A2 é, respectivamente, $4D, $45 e $53 — a notação hexadecimal dos códigos ASCII relativos a "M", "E" e "S".

Esse é um recurso importante, pois dispensa o programador da tarefa de converter mensagens e dados alfabéticos em listas de códigos ASCII, e também facilita a leitura da listagem. Além disso, sugere a possibilidade de se obterem saídas em tela a partir de programas em ASSEMBLY. Isso é relevante, pois até agora nos restringimos a armazenar resultados na memória e verificá-los usando programas especiais. No decorrer do curso, abordaremos a manipulação da tela, mas antes devemos estudar outros aspectos do ASSEMBLY. Contudo, não é difícil compreender co-

EXERCÍCIOS

1) A primeira rotina citada no texto utiliza o pseudocódigo DS para reservar $10 bytes de memória a partir do endereço representado pelo rótulo LABL1. Escreva um programa em ASSEMBLY para armazenar nesse bloco os números de $0F a $00, em ordem decrescente, um número por byte. Para tanto, use um loop e técnicas de endereçamento indexado; você necessitará das instruções DEX (decrementar o registrador X) ou DEC (IX + 0) (decrementar IX). O loop deve continuar decrementando o registrador de índice, mas sem chegar a ativar o flag zero. Use as instruções de desvio BNE ou JR NZ.

2) A partir do segundo programa citado no texto e utilizando as técnicas empregadas no exercício anterior, escreva um programa que copie a mensagem armazenada em LABL1, pelo pseudocódigo DB, num bloco de memória iniciando no endereço armazenado por DW, no rótulo LABL2. Se o endereço $98CE não for adequado ao seu computador, mude a inicialização; mas o programa deverá funcionar para qualquer endereço, e para mensagens de qualquer tamanho. Para tanto, seu programa deve usar o número de caracteres da mensagem como contador de um loop, ou então deve identificar o final da mensagem. Você pode colocar, por exemplo, um asterisco como caractere final de qualquer mensagem.

mo um programa endereça a tela, partindo do princípio de que sempre se armazenam resultados na memória, e de que a RAM de tela é também uma simples área de memória endereçável.

O recurso mais importante oferecido pelo pseudocódigo DB é que ele torna LABL1 semelhante a uma variável alfanumérica do BASIC. Quando escrevemos em BASIC

```
200 LET A$ = 'MESSAGE 1'
```

na realidade estamos criando um ponteiro que indica o início de uma tabela de bytes contendo os códigos ASCII de "M", "E", "S" etc. Cada vez que o interpretador BASIC encontra uma referência a A$, ele procura em sua própria tabela de símbolos o endereço apontado, isto é, a posição inicial do conteúdo de A$. Da mesma forma, em nosso programa em ASSEMBLY consideramos, LABL1 como equivalente de A$, pois já escrevemos uma rotina que manipula uma tabela por meio do endereçamento indexado.

Vemos assim que os pseudocódigos nos permitem eliminar dos programas os endereços e valores absolutos, substituindo-os por símbolos; isso torna as rotinas mais compatíveis e relocáveis. O que nos falta agora é um meio de acessar esses módulos transportáveis a partir do programa principal. Em outras palavras, precisamos de um equivalente do comando GOSUB do BASIC.

Esse equivalente existe: é a instrução JSR, na versão do 6502, ou CALL na do Z80. Ambas requerem como operando um endereço absoluto (que pode ser um rótulo), e colocam esse endereço em lugar do conteúdo do contador do programa. Portanto, a próxima instrução a ser executada será a primeira da sub-rotina assim endereçada. A execução continua, a partir desse ponto, até encontrar a instrução equivalente a RETURN — seja RTS ou RET. Esta irá substituir o conteúdo atual do contador do programa pelo endereço que lá estava imediatamente antes da chamada da sub-rotina por meio da instrução JSR ou CALL. Assim sendo, a próxima instrução a ser executada será aquela que vem logo em seguida a JSR ou CALL. Esse mecanismo é idêntico ao utilizado pelo interpretador BASIC ao executar uma sub-rotina por meio de um GOSUB, e dela retornar com um RETURN. Facilmente se entende isso tudo, mas surge o problema de como restaurar o conteúdo anterior do contador do programa após a execução do RETURN. A resposta é que as instruções JSR e CALL primeiro colocam o conteúdo do contador numa pilha, antes de substituí-lo pelo endereço da sub-rotina; por sua vez, a instrução RTS ou RET extrai esse endereço da pilha e o devolve ao contador. No próximo artigo do curso veremos em mais detalhe o que é essa pilha, e como a manipulamos.

Instruções essenciais

BEQ – DESVIO EM ZERO — 6502

Relativo F0 (2 bytes)

Desloca-se o conteúdo do contador do programa pelo valor do byte que se segue ao código de operação.

EFEITO SOBRE O PSR

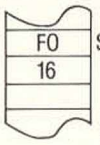
	S	V		B	D	I	Z	C
MSB								LSB

NENHUM EFEITO

Exemplo:

LOCALIZAÇÃO	CÓDIGO DE MÁQUINA	LINGUAGEM ASSEMBLY
8F00	F0 16	BEQ $16

	ANTES		DEPOIS	
Contador do programa	02	lo	18	
	8F	hi	8F	



JR Z – SALTO RELATIVO EM ZERO — Z80

Relativo 28 (2 bytes)

Desloca-se o conteúdo do contador do programa pelo valor do byte que se segue ao código de operação.

EFEITO SOBRE O PSR

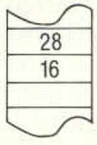
	S	Z	H		V	N	C
MSB							LSB

NENHUM EFEITO

Exemplo:

LOCALIZAÇÃO	CÓDIGO DE MÁQUINA	LINGUAGEM ASSEMBLY
8F00	28 16	JR Z,$16

	ANTES		DEPOIS	
Contador do programa	02	lo	18	
	8F	hi	8F	



INX – INCREMENTAR O REGISTRADOR X — 6502

Implícito E8 (1 byte)

Adiciona-se 1 ao conteúdo do registrador X.

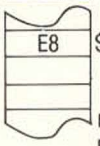
EFEITO SOBRE O PSR

	S	V		B	D	I	Z	C
MSB	X						X	LSB

Exemplo:

LOCALIZAÇÃO	CÓDIGO DE MÁQUINA	LINGUAGEM ASSEMBLY
F391	E8	INX

	ANTES	DEPOIS
PSR	????????	0??????1
X	FF	00



INC IX – INCREMENTAR IX — Z80

Implícito DD23 (2 bytes)

Adiciona-se 1 ao conteúdo de IX.

EFEITO SOBRE O PSR

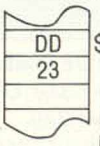
	S	Z	H		V	N	C
MSB							LSB

NENHUM EFEITO

Exemplo:

LOCALIZAÇÃO	CÓDIGO DE MÁQUINA	LINGUAGEM ASSEMBLY
F391	DD23	INC IX

	ANTES		DEPOIS	
Contador do programa	FF	lo	00	
	E7	hi	E8	



EDISA

Acompanhando a evolução
A Edisa desenvolveu uma linha de equipamentos capazes de se adaptarem à evolução das empresas, como o multiusuário/multitarefa ED-281

A Edisa é uma empresa voltada estritamente para computadores tanto de médio como de grande porte. Seu ED-680 é o primeiro equipamento nacional a empregar o processador Motorola 68000.

Como quase todas as empresas nacionais de computadores, a Edisa - Eletrônica Digital S.A. também teve início na década de 70. Por intermédio de um contrato de cooperação técnica com a empresa Fujitsu, do Japão, a Edisa começou sua produção de minicomputadores no país em novembro de 1977. A partir daí, sua preocupação tem sido desenvolver e aperfeiçoar equipamentos para atender à crescente automação industrial, comercial e, mais recentemente, bancária. Atualmente, além de sua linha de minicomputadores, a Edisa produz também micros e supermicrocomputadores.

Um ano após a fundação da empresa, seu controle acionário passou às mãos do grupo financeiro Iochpe, tradicional no setor bancário, mas que também atua nas áreas agrícola e de papel. Em 1980, a Edisa começou a fabricar os equipamentos ED-300 — até então apenas montados no país. Nesse mesmo ano teve início a produção dos sistemas de entrada de dados ED-100. Em 1981, a empresa conseguiu mais uma fatia do mercado, passando a fazer os terminais financeiros ED-3400, destinados à automação bancária.

Na área de recursos humanos, a Edisa possui um setor de desenvolvimento destinado a dar oportunidade aos novos profissionais de informática e engenharia.

Esse apoio à educação é dado também fora da companhia, através de acordos com universidades, porém mais voltado para a área de software, onde os alunos desenvolvem programas subsidiados pela empresa. A produção de software decorre paralela à criação de novos equipamentos na empresa, possibilitando a atuação permanente de estudantes do ramo.

Os protocolos — ou periféricos destinados à comunicação entre dois ou mais tipos de processadores — também despertam interesse nas universidades. Assim, as escolas desenvolvem um projeto de periférico e o apresentam à empresa. Se houver interesse em sua comercialização, a Edisa firma contrato com a universidade e investe no desenvolvimento do aparelho.

Muitas escolas superiores possuem bons laboratórios e técnicos altamente qualificados, dispondo-se a desenvolver programas para a Edisa. Entre as universidades contratadas pela empresa estão a Federal do Rio Grande do Sul e a Federal do Rio de Janeiro.

A fábrica da Edisa localiza-se no Distrito Industrial de Gravataí, Rio Grande do Sul, numa área de 36.000 m². A empresa possui ainda filiais nas cidades de Porto Alegre, São Paulo, Rio de Janeiro, Brasília, além de escritórios nas principais cidades brasileiras.

A Edisa preocupa-se com a tecnologia avançada dos equipamentos estrangeiros. Exemplo disso é o ED-680, primeiro supermicrocomputador nacional a utilizar o processador Motorola 68000, que vem sendo usado há apenas dois anos no mercado internacional.

Outra característica da empresa consiste na atualização versátil de seus equipamentos, de forma a não tornar obsoletos os aparelhos adquiridos por seus clientes. Essas mudanças também estão presentes no ED-680, em sua nova versão, que utiliza discos Winchester com capacidade de armazenamento de 72 Mbytes por unidade, embutidos no próprio gabinete da CPU e acondicionados em espaço magnético selado.

A Edisa produz também o ED-381, um minicomputador que pode ser conectado a equipamentos de grande porte da linha IBM ou Burroughs, ou a equipamentos que utilizam as linguagens universais. Com memória real de 24 Kbytes a 256 Kbytes, esse míni desempenha funções em lotes ou on line.

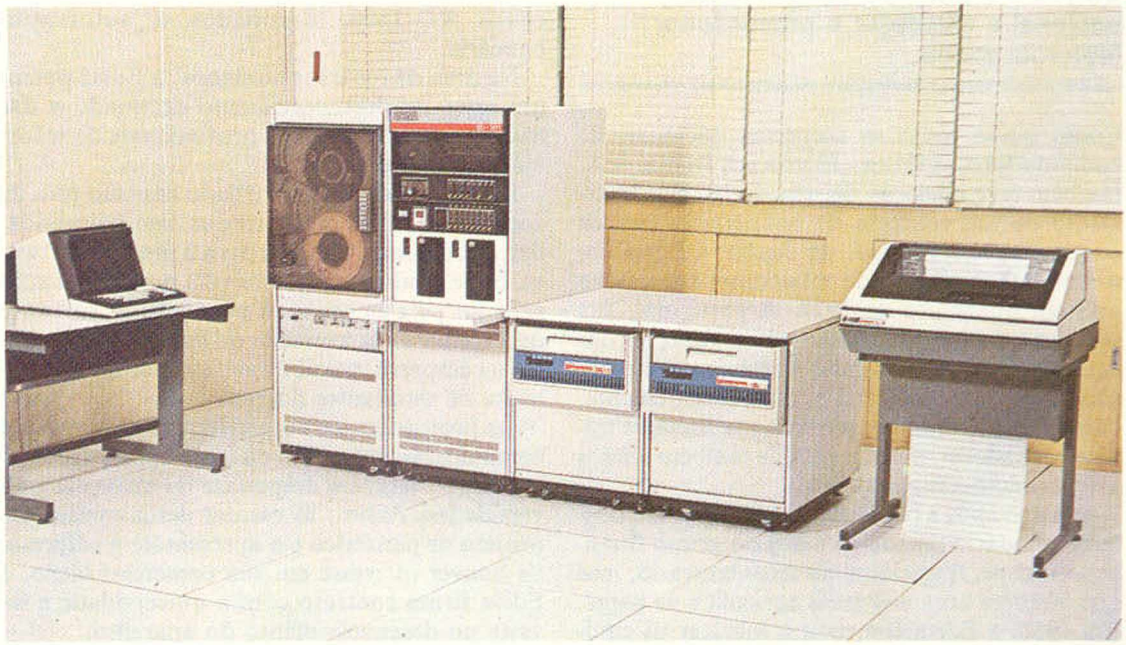
A área comercial é a mais cobiçada pelas indústrias de informática. A cada dia surgem novos modelos de equipamentos versáteis, projetados de forma a se adaptarem à evolução da empresa, ou sistemas práticos para automatização de firmas comerciais em curto espaço de tempo. Voltada para esse campo, a Edisa criou a linha ED-200. O multiusuário/multitarefa ED-281 tem a capacidade de incrementação de acordo com o crescimento da empresa. Permite que várias pessoas executem diversas tarefas ao mesmo tempo, nos múltiplos setores da companhia. O ED-251 trabalha em 8 ou 16 bits. É um microcomputador desk-top, com processador Z80A e CPU com 64 Kbytes. Utiliza discos flexíveis de 5 1/4 polegadas.

A Edisa também está voltada ao desenvolvimento de softwares para suprir seus equipamentos. Além dos programas elaborados pelas universidades, a empresa dispõe de pessoal especializado para atender a esse mercado, além de prestar serviços de consultoria aos clientes, a fim de orientá-los na aquisição dos sistemas mais adequados a suas atividades.

Mesmo os aplicativos desenvolvidos pela empresa possuem flexibilidade para serem adaptados às necessidades de cada cliente e ao perfil de sua empresa. A série ED-800 é uma família de sistemas interativos compatíveis entre si desenvolvida pela Edisa. O modelo ED-848 atende às necessidades de uma empresa de médio porte, mas tem possibilidade de expansão. Opera também como nodo principal em uma rede de sistemas distribuídos, através do software de comunicação de dados. Comporta até 152 usuários e pode ser processado em lotes ou on line.

O sistema ED-868 é um pouco mais sofisticado, dirigido a empresas de maior porte. Tem configuração de memória de 4 Mbytes, podendo expandir-se até 16 Mbytes. Apresenta capacidade para atender às necessidades de processamento centralizado ou distribuído para até quatrocentos usuários.

A Edisa oferece treinamento a seus clientes para possibilitar maior intimidade com os equipamentos. Esse treinamento também é oferecido aos funcionários das assistências técnicas autorizadas para manutenção dos equipamentos.



Solução de porte
O minicomputador ED-381 pode ser conectado a computadores de grande porte ou a qualquer outro que utilize linguagens universais.

mória, reservados ao armazenamento dos dados usados para recriar as frases necessárias. O Flyerfox usa o método de codificação "preditivo linear". Nesse sistema, as palavras são convertidas em sinais digitais, posteriormente armazenados na RAM. Quando se requer determinada palavra, os dados digitais correspondentes são acessados, reproduzindo-a por intermédio do chip SID do Commodore.

Os gráficos do jogo são todos de alta resolução. Os quadros de vídeo consistem em vistas frontais, mostrando o painel de instrumentos e o céu através da cúpula transparente da cabine de comando. O programa oferece vários instrumentos de auxílio ao vôo, incluindo bússola e radar. Este detecta a aproximação dos caças Mig, o que dá ao jogador tempo de se preparar para o combate. Além disso, duas luzes intermitentes, uma de cada lado do horizonte artificial, indicam ao jogador se os Mig estão acima ou abaixo do ângulo de visão.

O desenrolar dos combates é rápido e bastante realista. Quando um Mig surge na tela, o programa produz um bip de advertência. O jogador deve, então, manobrar seu aparelho de forma a tentar enquadrar o atacante na mira. Isso não é fácil, porque os aviões se evadem e mergulham a grande velocidade. Quando conseguir pôr o alvo na mira, o jogador poderá disparar seus mísseis, dirigidos pelo calor das turbinas do inimigo; mas eles não são infalíveis e freqüentemente os caças inimigos escapam.

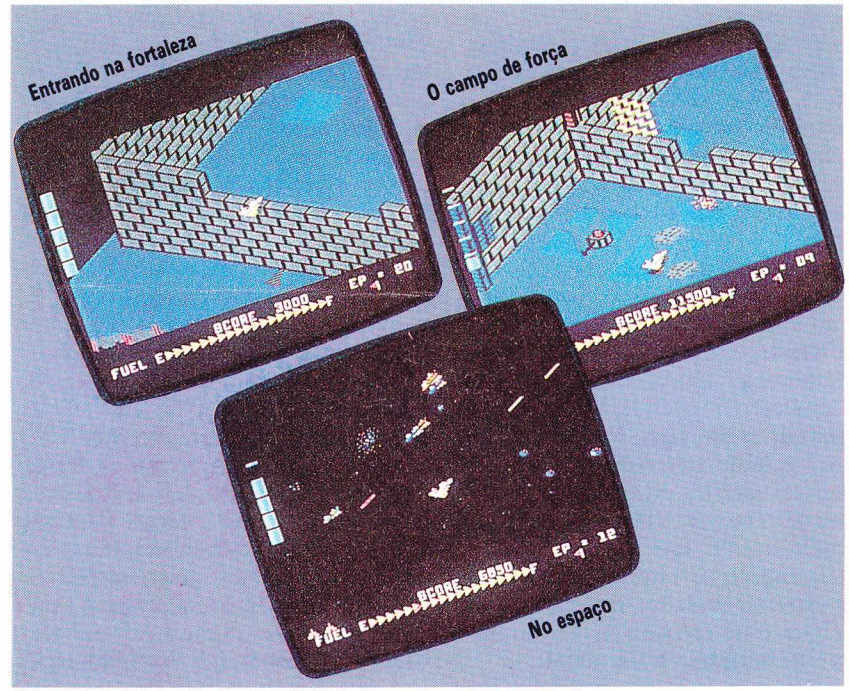
Embora os gráficos sejam de alta qualidade, não apresentam muita variação. A ilusão de movimento é dada pela alteração nos padrões das nuvens, e o solo consiste simplesmente num quadriculado estilizado. Deve-se dizer também que o próprio avião escoltado pouco acrescenta ao jogo. Deixando de lado o paralelo óbvio com o caso do Jumbo coreano (da Korean Air Lines), o que mantém em aberto uma acusação de mau gosto, fica difícil entender por que se incluiu tal avião se ele apenas pode ser visto pela traseira. Além disso, o Flyerfox é incapaz de alcançá-lo após entrar em combate com os caças inimigos. Ao contrário de um avião real, o Jumbo nem mesmo esboça qualquer ação evasiva ao ser atacado.

O Flyerfox representa uma nova tendência dos jogos para computadores. A síntese de voz inserida em programa é um tema já considerado há algum tempo. O jogo da Tymac usa a fala, mas não requer interface especial ou periférico para hardware. Daí constituir um verdadeiro marco no desenvolvimento dos jogos para computadores.

Flyerfox: Para os computadores Commodore 64, Atari e Apple II.
Editor: Tymac Corporation.
Autores: Gregory Carbonaro, Charles Teufert, Ronald Pintus e Arthur Aspromatis.
Joystick: Necessário.
Formato: Disco ou cassete.

FORTALEZA ESPACIAL

ZAXXON (análise)



A fronteira final
Três fases diferentes do Zaxxon. O jogador inicia pilotando a espaçonave na primeira das duas fortalezas. Depois de vencê-la, a astronave retorna ao espaço, onde deve eliminar, ou evitar, a esquadrilha inimiga. Finalmente, só depois de superar as defesas da segunda fortaleza, enfrenta o próprio robô Zaxxon.

Os jogos prontos para microcomputadores dependem muito, para sua inspiração, dos fliperamas: grande número de bem-sucedidos pacotes de software são adaptações de temas consagrados em fliperamas, como o Zaxxon, programa hoje disponível para várias linhas de micros.

O objetivo do jogo é pilotar uma espaçonave por entre vários obstáculos, numa tentativa de destruir o robô Zaxxon. No início, ela se encontra nas profundezas do espaço; posteriormente, uma fortaleza espacial, cercada por altas muralhas, surge à sua frente.

Para avançar, a espaçonave deve passar através de uma abertura nessa muralha, antes de se defrontar com o primeiro dos muitos obstáculos que tentam barrar-lhe o caminho: mísseis, antenas parabólicas de radar e depósitos de combustível — tudo desenhado realisticamente em três dimensões.

Os depósitos de combustível adquirem singular importância, uma vez que sua destruição faz aumentar o combustível nos tanques da espaçonave.

Os gráficos do jogo são muito atraentes. Usam-se três diferentes sprites para a espaçonave: um para mergulho, outro para subida e um terceiro para vôo normal. Conforme ela sobe ou mergulha, esses sprites aumentam ou diminuem seu tamanho de forma convincente.

Os obstáculos da superfície também são bem desenhados, e a perspectiva confere muito realismo ao jogo. Um toque particularmente interessante é a sombra da espaçonave, que aumenta quando esta se aproxima do solo, passando por sobre os objetos na superfície.

A destruição da espaçonave pelo fogo inimigo nessa fase faz o jogo retornar ao início. Mas, depois de praticar o suficiente, o jogador verá que consegue superar a maioria dos riscos. É válido o esforço no sentido de destruir as antenas de radar, pois, se deixadas intactas, guiarão mísseis em trajetória horizontal contra a espaçonave — e cada um deles só é destruído com três impactos diretos.

A avaliação correta da altitude requer extremo cuidado: um erro de cálculo tem como conseqüência ir diretamente de encontro a um míssil.

Às vezes o jogador se defrontará com um campo de força elétrico. Deverá então subir bastante para passar acima dele e, logo em seguida, mergulhar para um nível baixo, onde possa eliminar as casamatas inimigas, concretadas na superfície e que estarão à sua espera. É aconselhável livrar-se de tantas quanto possível, pois cada uma destruída nesta etapa significa uma a menos para eliminar mais tarde.

Depois de atacar com sucesso as defesas da fortaleza, a astronave ganha novamente o espaço, onde se defronta com uma esquadrilha inimiga de ataque. Nesse ponto, o jogador enfrenta o problema de disputar um jogo em três dimensões numa tela bidimensional.

Para atacar a nave inimiga deve-se subir ou descer ao nível correto, o que envolve tentativas e erros, visto que a única indicação dada é o tamanho relativo das várias naves: a espaçonave encontra-se em posição de tiro quando tem aproximadamente o mesmo tamanho da nave inimiga. Um dedo rápido no botão do joystick constitui vantagem decisiva nessa fase do jogo. Deixando novamente o espaço, a astronave retorna à fortaleza espacial. Só que, dessa vez, a abertura na muralha está menor e os alvos bem mais difíceis de atingir. É agora que o jogador vai enfrentar o robô Zaxxon, que avança por uma grade ao seu encontro.

Nesse ponto, a espaçonave pára totalmente e sua única defesa é destruir o robô antes que ele a pegue. Para tanto, o jogador precisa conseguir três impactos diretos nos lança-mísseis do monstro. Enquanto isso, Zaxxon avança inexorável; e, se conseguir atingir a nave, o jogador terá de voltar ao início.

Zaxxon: Para os computadores Sinclair Spectrum, Commodore 64, Atari e Apple II.
Editor: US Gold, Inglaterra.
Autor: Peter Adams.
Joystick: Necessário.
Formato: Disco ou cassete.

ESCOLTA AÉREA

Flyerfox (análise)

Os Mig atacam

Busca e destruição
O piloto do Flyerfox obtém as informações sobre a aproximação dos caças inimigos tanto pelo painel de instrumentos como pela observação visual do céu. Os pontos na tela do radar são aviões, embora nem todos ataquem. Dois quadrados brancos de cada lado do altímetro indicam a altitude relativa dos Mig, enquanto a bússola ajuda a relocalizar o Jumbo.

Embora os jogos para computadores tenham percorrido um longo caminho desde os primeiros dias do Space Invaders, a maioria dos progressos ocorreu no desenvolvimento de gráficos. Os programadores concentraram-se principalmente na busca de novas formas de introduzir dados, de um número cada vez maior de páginas gráficas, numa quantidade limitada da RAM. Nesse meio tempo, as excelentes possibilidades sonoras de muitos microcomputadores permaneceram ignoradas.

Uma empresa americana, a Tymac, começou a distribuir uma série de jogos incorporando síntese vocal, mas sem realmente usar uma interface. O primeiro desses programas é o Flyerfox, que oferece aos proprietários do Commodore 64 a oportunidade de entrar em combate feroz com caças inimigos. Os programas de simulação de vôo são bastante comuns, mas este se destaca por acrescentar rotinas de fala incorporadas que dispensam hardware adicional. No Flyerfox, o jogador pilota um caça que escolta um Jumbo levando altos funcionários do governo a bordo, voando num espaço aéreo em litígio. Caças Mig inimigos tentarão derrubar o Jumbo, e o objetivo do jogador é fazer contato com eles e destruí-los. A síntese vocal usada no jogo envolve uma série de mensagens transmitidas do Jumbo para o caça de escolta.

O sintetizador de voz é parte do software e ocupa cerca de 11 Kbytes de me-